

Inside: The Ultimate OS/2 Game

OS/2 Monthly

The Independent Guide to OS/2 Computing

Issue Six / \$4.50

Beginner's Slope

Plus

OS/2 & MIS
Containers
The Dos Environment



PLEASE DISPLAY THIS MAGAZINE
UNTIL FEBRUARY 15, 1993

GPF Review





The Only Tool ... to create, modify, You'll Ever Need manage, browse and

translate resources such as dialog boxes, bitmaps, icons, cursors and menus.

Increase the productivity of your development work by organizing resources into projects and seamlessly integrate all the resource editors.

Reduce the amount of time required to design the user interface of your application—use the state of the art visual editors to interactively design and modify dialogs, bitmaps, icons, cursors, menus, accelerators, string tables and all other resources—or just use any of the hundreds of professionally designed resources included.

Capture images from anywhere on the screen directly into the image editor. Create bitmaps, icons and cursors from the captured image.

Customize the look and feel of your applications—directly extract, modify or replace any resource in any program file, EXE, DLL—no source code is needed.



Translate applications into
foreign languages without

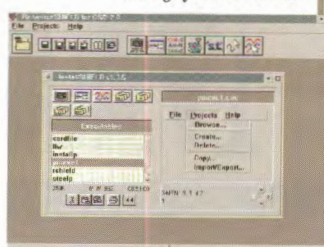
having access to the applications source code—

directly modify all messages, menus and dialogs into any national language.

Edit, copy and paste resources directly into and from any EXE, RES, DLL, RC program file. Eliminate the time consuming edit-compile-link cycle—increasing your productivity dramatically.

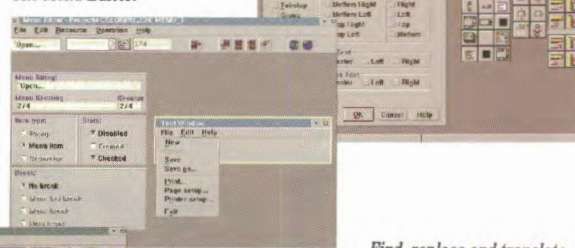
Migrate your applications to new 32-bit OS/2 2.0 and NT applications utilizing the transparent and automatic resource translation capabilities. Create your resources once on any platform and instantly use them on any other platform.

Project views and preview
browsers allow you to efficiently
manage your resources.

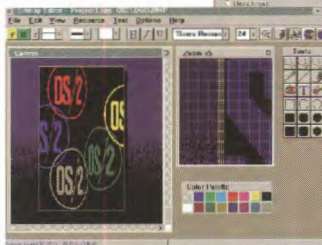


Quickly design complex
dialogs using powerful
design tools in the
Dialog Editor.

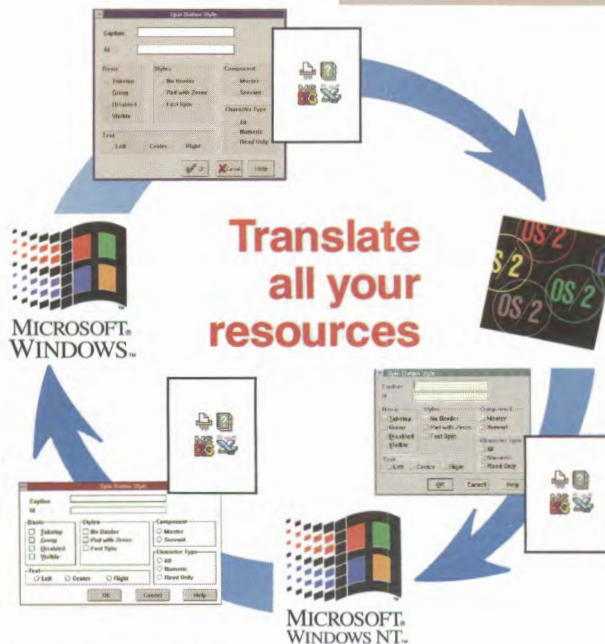
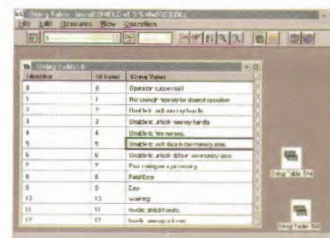
The Menu Editor.



Create bitmaps, icons,
cursors using the
advanced image editors.



Find, replace and translate
text messages in string tables.



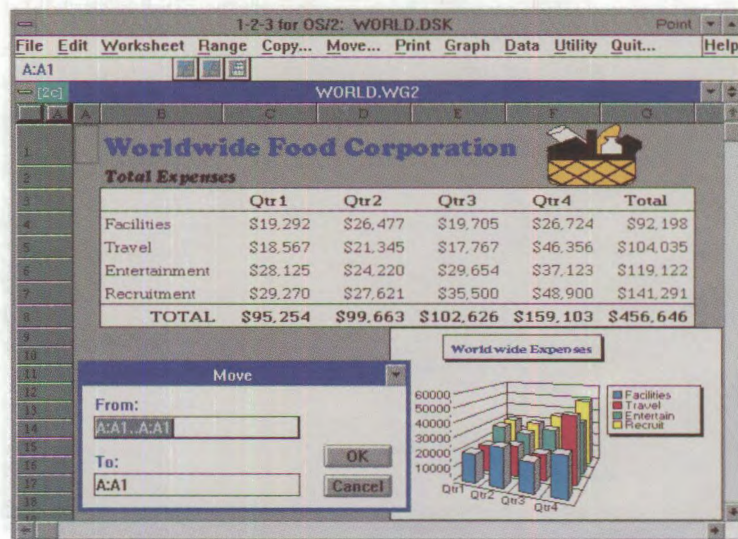
The Stirling Group
Making it easy to make™

172 OLD MILL DRIVE • SCHAUMBURG, IL 60193 • USA **1-800-3 SHIELD**
1-800-374-4353

CALL 708-307-9197 • FAX 708-307-9340
CompuServe: GO STIRLING • MCI Mail: STIRLING

Getting the most out of OS/2 is as easy as 1-2-3.

Nothing exploits the full graphical environment of OS/2® like Lotus® 1-2-3® for OS/2. It's the only spreadsheet that takes full advantage of the speed, memory and multi-tasking capabilities of OS/2, while offering all the power and functionality you've come to expect from 1-2-3. This includes Solver and Backsolver, true 3D worksheets, file-linking and access to external databases. As well as such industry-leading features as WYSIWYG (What-You-



Thanks to its multi-windowing capabilities, 1-2-3 for OS/2 lets you look at your graphs and spreadsheet simultaneously. Whether it's working with a 3D file or bringing in information from external databases, the full power of 1-2-3 for OS/2 is always within reach.



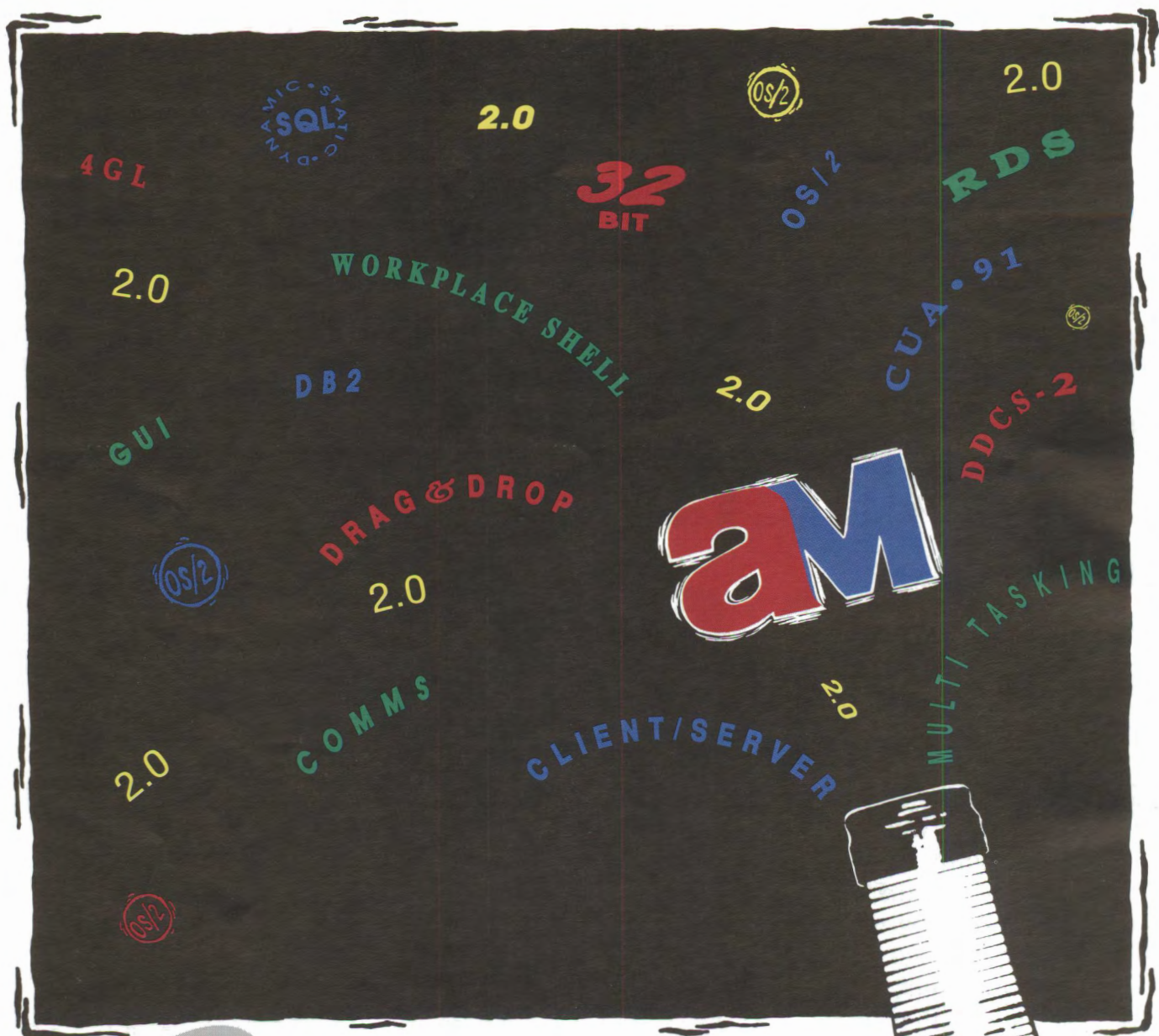
See-Is-What-You-Get) and Collections, which allows you to work with multiple cells and ranges simultaneously.

1-2-3 for OS/2 is also the only spread-

sheet that's fully compatible with IBM® OS/2 version 2.0. And of course, it's compatible with all previous versions of 1-2-3, across all platforms.

For a free demo disk, including sample spreadsheets to show you just how powerful 1-2-3 for OS/2 is, call **1-800-TRADEUP, Ext. 6565.**

1-2-3 for OS/2



CELEBRATE THE POWER

AM(Applications Manager)...the only OS/2 client/server development tool providing 32-bit applications exploiting the Workplace Shell and accessing data from local LAN or remote host databases. **AM** increases productivity at least 10-fold compared to conventional application development languages. **AM** is currently in use by over 1,000 organizations worldwide and is the leader in the delivery of proven, live applications.

You too can experience the power...**CALL NOW...508-640-1080.**

INTELLIGENT ENVIRONMENTS INC., 2 HIGHWOOD DRIVE, TEWSBURY, MA. ALL PRODUCTS AND PRODUCT NAMES MENTIONED ARE REGISTERED TRADEMARKS OF THEIR RESPECTIVE COMPANIES.


CONTENTS

FEATURES

THE OS/2 2.0 DOS ENVIRONMENT PART III

Les Bell

7

ON THE EVE OF INSTALLATION

Bill Zinsmeyer

21

OS/2 + MIS

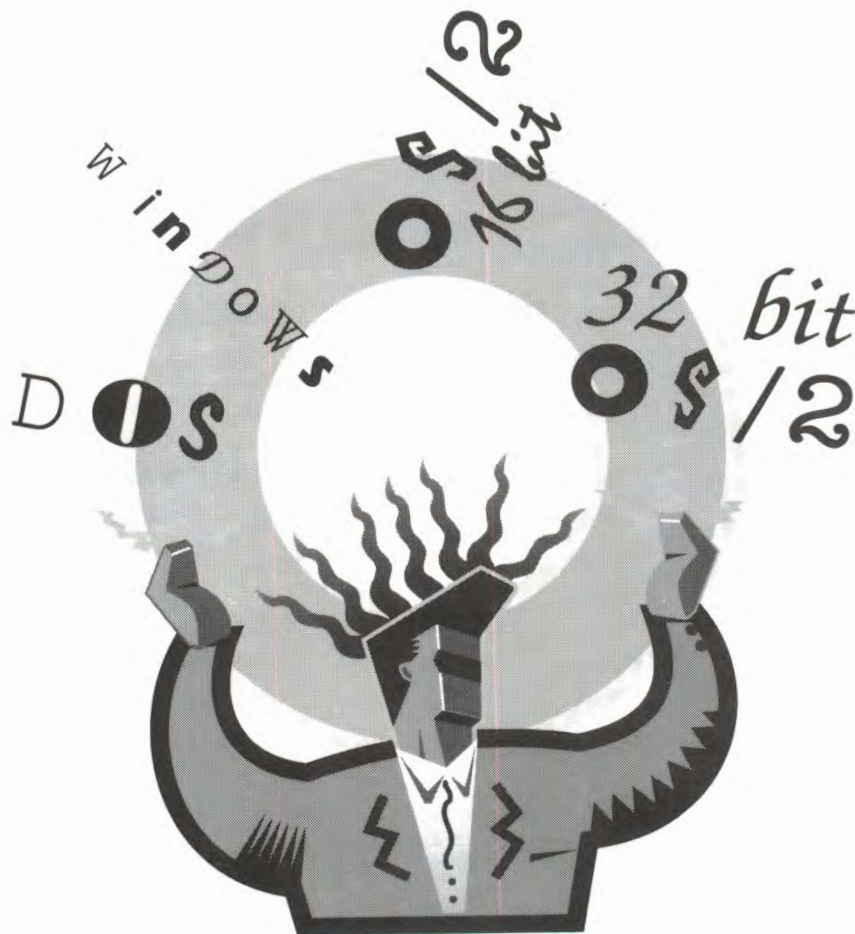
Gary Murphy

28

DEPARTMENTS

PUBLISHER'S NOTES	3	FOR IMMEDIATE RELEASE	30
EDITOR'S CORNER	4	ADVANCED PM PROGRAMMING	32
Q&A	5	Introducing Containers . . . Guy Scharf	
David Hunt		THE ULTIMATE OS/2 GAME	43
DISCOVERING THE WORKPLACE SHELL	13	Paint By Numbers . . . Timur Tabi	
Brett Kotch		OBJECT OBJECTIVE	45
REVIEW	15	Building Programs . . . David Moskowitz	
Ron Beauchemin & Paul Duncanson		IN THE TRENCHES	48
		Eric Pinnel	

In order to help you juggle all the possibilities of OS/2



Subscribe to OS/2 Monthly

1 Year subscription—12 issues / \$39

2 year subscription—24 issues / \$75

Foreign subscriptions must be paid in US dollars.

\$45 / year to Canada or Mexico. \$77 / year to all other countries

For fastest service call 1-800-365-2642

or

send your name, address, city, state, and zip code along with your
Check/MO or Visa or Mastercard number with exp. date and signature to

JDS Publishing, P.O. Box 4351, Highland Park, NJ 08904

PUBLISHER'S NOTES

Publisher

Joel Siragher

Editor

Brett Kotch

Contributing Editors

Les Bell, Bert Moshier,
David Moskowitz, Eric Pinnel, Guy
Scharf, Timur Tabi, Ron Beauchemin

Technical Review Board

Noel Bergman, Larry Finkelstein
Mel Hallerman

Graphical Director

Deborah S. Weightman

Volume 1, Issue 6, December 1992

OS/2 Monthly is published monthly by
JDS Publishing, P.O. Box 4351, Highland
Park, NJ 08904, 908-246-0952.

OS/2 Monthly is published
independently of IBM Corp., which is not
responsible in any way for editorial policy
or other content of this publication.

Postmaster: There is a second class
subscriber mailing permit pending. The
original entry point is the Kilmer Mailing
facility in New Brunswick, NJ. Address
corrections requested.

Subscription rate for the U.S. is \$39. for
12 issues. Canadian and Mexican are \$45.
All other foreign subscriptions are \$77.
Visa/Mastercard accepted.

OS/2 Monthly welcomes unsolicited
submission of written material and
artwork; however, no responsibility for
such submissions can be assumed by the
publisher in any event.

OS/2 Monthly is a trademark of JDS
Publishing. All rights to contents of this
publication are reserved, and nothing may
be reproduced from it in whole or part
without first obtaining permission in
writing from the publisher.

Articles and items appearing in OS/2
Monthly do not necessarily reflect the
views and opinion of JDS Publishing, its
staff, or organization.

All trademarks used herein are property
of their respective holders.

OS/2 is a registered trademark of IBM
Corporation and is used by JDS Publishing
under license.

We want to thank all our readers for their patience in waiting longer than usual for this sixth issue of OS/2 MONTHLY. We have switched printers and reorganized our editorial staff in an effort to keep bringing you the best possible articles, columns and features dealing with OS/2. Your subscription will be extended to include the full year's complement of magazines.

This month, we are adding a new column, "Beginners' Slope" by Bill Zinsmeyer. As the title would indicate, this column will approach OS/2 from a beginner's level and will soon have you off the bunny slope.

We're also happy to announce that next month we will be featuring a new column which will explain the "ins" and "outs" of REXX.

If you possess a particular area of expertise and would like to contribute an article to OS/2 Monthly, please submit an outline of your idea and a sample of any previously published writing to:

JDS Publishing
P.O. Box 4351
Highland Park, NJ 08904

Joel Siragher

EDITOR'S CORNER

I believe that we are all in agreement when I claim that with regards to version numbers, $2.0 > 3.1$. But yet, if we were to multiply each by one million we arrive at the following contradiction:

$$(1,000,000) * 2.0 < (1,000,000) * 3.1.$$

IBM has sold one million shrink wrapped copies of OS/2 and comparing this to MicroSoft's first million, there is a profound difference in the impact that this number has had. Contrary to popular thought, the above contradiction isn't based entirely upon the efficiency and ability of the MicroSoft marketing machine vs. IBM's or solely on the technical merit of Windows.

When the success of Windows is discussed, there is a small distinction that is frequently overlooked. MicroSoft (MS) would want you to believe that with the release of 3.1, they single-handedly revamped the personal computing market. Actually, it wasn't until 3.1 became available that the personal computing market had the opportunity to revamp itself.

In the beginning of the silicon era, change was fast, people packed up their belongings and moved to the frontiers with trepidation, needing to keep abreast of all of the latest changes. Magazines focusing on personal computers kept us informed and kept happy.

After several years, the market grew to enormous proportions. Killer apps were few and far between and the value of the mammoth computer magazines decreased. People were saturated with information and content with the knowledge that their grey matter now contained. Change wasn't as radical and a constant stream of new information was not as necessary as it had been. Reporting on computers had become stale. Change was necessary and Windows provided the trade magazines with an opportunity to convince the public that their magazines were needed.

Windows represented change, it offered magazines a new avenue to seduce the reader to continue subscribing. All of the major areas of computing had been defined, so it was time to change how we interacted with computers. This is where the genius of MS lies. They saw the decline in the computing magazine market and

revived it. This would explain the overbearing prejudice of Windows over OS/2.

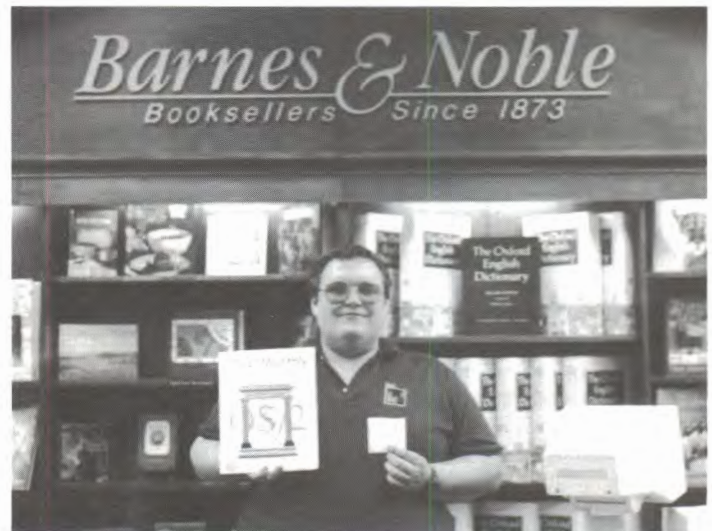
Possibly one of the most honest statements I have read, is the general atmosphere about OS/2 supporters, and possibly why we are so zealous at times, is from Mark Minasi, which originally appeared in the October issue of COMPUTE:

"Despite my personal feeling that OS/2's success would be a very positive thing for the industry, you can't have missed the somewhat defensive tone of this article. The defensiveness arises mainly because, while there are reasons to like OS/2 as well as reasons to dislike it, I fear that many won't even give it a try because of the very uninformed press this operating system has received."

This obviously puts IBM in a difficult position. Even with fighting the wave of prejudice IBM is having some success and new end-users are entering the arena. As a result, we at OS/2 Monthly, have had many requests for more material focused at the newcomer to OS/2. We have gone back to the beginning. Bill Zinsmeyer alleviates the installation blues for OS/2, in his new column, which for its introduction is our cover article.

AND THE WINNER IS . . .

The OS/2 Golf T-Shirt goes to Bert Moshier for being the first person to spot OS/2 Monthly in a store. Congratulations Bert!



Bert Moshier

BRETT KOTCH

Q&A

DAVID HUNT

David Hunt is the Manager of R&D for a small Midwest computer service bureau. One of his current duties is the supervision of an OS/2 development group. He is anxiously awaiting your questions and can be reached on CompuServe as 76326.3641 or by letter care of editor at the address in the masthead

I did something a little bit unusual for me a couple of weeks ago, I went on a sales call. Generally speaking, I couldn't sell anything to save my soul. If I depended on a sales job to provide for my family we'd probably be among the ranks of the homeless.

Over the past thirty years I've discovered two exceptions to that rule. Sometime during my college years I realized that I could sell things that I truly believed in and enjoyed. One of those things is bicycles. My most enjoyable college job was as a mechanic and sometime salesperson in a bike shop.

I can't explain the joy that I get just from seeing a beautiful bike. And to ride one is almost ecstasy. One of the best times I had at the recent Windows & OS/2 Conference in Boston was when I wandered into a bike shop and spent an hour just hanging out.

The second thing that I discovered I could sell during college was cars — not the standard used "sleds" that you see littering the ever-present "Honest John's", or the bland clones that Detroit and the Japanese peddle, but truly breathtaking true automobiles such as classic sedans or sports cars. Unfortunately, most of those automobiles have been snatched up by people who have too much money and not enough time. Consequently, prices have skyrocketed and the beloved beasts sit in garages waiting to be exercised by an owner who never appears. I could sell the darn things, but I sure can't afford to buy one.

So what's the point? I've discovered some-

thing else I can sell — OS/2! A friend of mine sells computers in the retail channel. I don't see how any of them make money, but he's got a house, and his family seems to have food and clothing. A couple of weeks ago he called to ask a favor. One of his old accounts had been turned over to another rep at the store and had been handled badly. In fact the account (a rather large one) was planning to walk. Management fired the other rep and gave the account back to my friend. He was trying to salvage whatever he could from a bad situation when he was asked about OS/2.

At this point my friend had a problem. He was trying to reestablish his firm as a knowledgeable vendor, but nobody there knew anything about OS/2. (At this point I could mention that I'd been trying to get them to support OS/2 for a year or so, but that is probably another topic.) So he called me to ask a favor. "Sure" I said, "I'd be happy to go on a sales call with you." It was actually fun.

We visited the network support group for a large hospital. My friend had convinced them to give his firm one chance to make things right and had just sold them six or eight IBM PS/2 computers. These machines were going to come preloaded with OS/2 2.0. What the support folks wanted to know was "do we want to keep OS/2 on these machines, or should we reformat the drives and load DOS (and maybe Windows) on them?" Having been responsible for supporting a medium-sized network at one time, I could understand where they were coming from. Did it make any sense for them to accept and support one more operating system?

In talking with them, several things came out. Ninety-nine percent of their people were using DOS only. A couple of folks around the hospital had experimented with Windows (I assume that none of them had inhaled.) The only one in the whole organization who actually used Windows on a regular basis was their boss, and he thought DOS and Windows were the way to go.

I pointed out that I wasn't there to sell them on OS/2. My only role was to answer their questions about the operating system. I did make it clear that I was an OS/2 user and that my job involved the supervision of a fairly ambitious OS/2 development project. I explained to them that as



See your entire system graphically at a glance.

NEW OS/2 Tree utility is a super Programmer, LAN Administrator Productivity Tool.

- Complete programmability every function/key can be customized!
- Display your tree structure for all system and LAN disks concurrently.
- Manipulate your files, directories, entire disks and/or entire system!
- HPFS fully supported
- Pop-up list of files for any directory
- Edit/browse file(s) using any EDITOR!
- Extensive FILE Search and TREE directory search capabilities
- Custom task manager
- Custom document formatting and printing
- Multiple pop-up to-do lists
- Date and Appointment reminder
- Display system clock in any application
- Upload/download files between mainframe and PC
- Custom programming language for macros and batch operations
- 30 day unconditional guarantee.

Intro Price: \$49.99
\$249.99 Professional Version
Network Licenses Available

Levine Computer Consulting Services
2501 Porter St.
Suite #838
Washington, DC 20008

Orders / Fax / Inquiries / Helpline
(202) 966-5309

a result I felt fairly comfortable posing as an OS/2 expert. One other noteworthy item, just to make the story semi-believable: the company my friend works for is loosely affiliated with the one I work for, so everyone was comfortable that I did indeed represent some in-house expertise for the retailer.

The questions came fast and furious. How much disk space does OS/2 need? How much memory would their workstations need to run it? Are there applications available to run on OS/2? Is OS/2 hard to use and support? Would OS/2 run on their clones?

Slowly and surely, I worked through their onslaught. To install the entire OS/2 operating system, I would want at least 40MB of free space on the drive. This would allow for the entire system (full hardware support, online help, applets, etc.) and would leave room for an effective swap file. I've installed OS/2 on 30MB drives without much trouble, but I'm not sure I'd want to run it on anything less. By the time you add comparable functionality to DOS or Windows, you are frequently nearing this disk volume and the complexity of the configuration will be much worse.

I mentioned that initially IBM had said that systems would need 4MB to run OS/2. I've run on systems with 4MB and it isn't enjoyable. Even IBM is now saying 6MB is a better minimum. Most anybody else you talk with is saying 8MB. Memory is cheap enough now that I try to get 16MB on all the systems I order for my department. I don't have anything less than 12MB, though. You just can't justify giving up the performance improvement for the minimal dollars that an extra 4MB to 8MB of RAM would save you. These memory requirements are certainly greater than what DOS needs (or can use). Windows will run (I've been told) on 2MB systems. To get it to run well, though, you should consider 6MB to 8MB. This isn't much different than OS/2.

There are OS/2 applications available to do just about anything you would want to do on a computer. The one caveat is that they may not be the ones that you are used to using under DOS. On the other hand, just about everybody who is anybody has a Windows version of their software out now. The one thing that OS/2 will do very well though (better than Windows) is run your current DOS applications. In explaining that all of the current DOS applications they are supporting could be running (not just ap-

pearing) in windowed sessions on an OS/2 desktop I hit upon something that was really worth something to the folks at this hospital. The icing on the cake was when I told them that each of these sessions could be individually configured. Things like "unlimited" EMS or XMS memory for DOS applications that are currently causing unbearable configuration and support problems on their DOS workstations went a long way toward selling the folks in the room that afternoon.

We talked about ease-of-use. I mentioned an in-house project our group had just completed. We wrote an application for a group of users in our company who are dyed-in-the-wool computerphobes, and they love it! The interface is intuitive, easy to learn and easy to use in their day-to-day work environment. Our in-house support team has found that support time has been cut. In short, the interface has been a resounding success.

Would OS/2 run on their clones? This I could answer with absolute certainty, Yes! I knew it would because my company had bought the same clones that they had in the hospital.

At the end of the meeting they were all interested and I invited them to stop by my shop for a visit. They were up two days later with my friend. I demonstrated many of the DOS applications they were supporting running on my OS/2 desktop. I showed them an OS/2 communication program downloading messages from CompuServe at the same time that I had a DOS communication program downloading files from a BBS through our network modem pool, both in the background! My group showed them some of the neat "gee-whiz" stuff and multi-media gizmos that we play with in our development projects. Finally, we went to visit the group of computerphobes who are using OS/2 in their day-to-day work. At the end of the visit the folks from the hospital (and my friend) were thrilled.

The next day my friend called. He'd decided to install OS/2 on his computer and said that maybe it was something he should be using. I'd managed to sell OS/2 to him.

I haven't talked to him for a month now — we're both too busy most of the time to keep in touch on a regular basis. I think that maybe tomorrow, I'll take a few minutes to track him down. I'd like to find out what the folks from the hospital have decided about OS/2. After all, how else can I know how good an OS/2 salesman I am? ♦

THE OS/2 2.0 DOS ENVIRONMENT

PART III

**LESS
BELL**

Les continues his discussion on the DOS environment in OS/2 2.0 — Editor

OS_STARTUP_DRIVE

See the section on Running Real DOS, below. This parameter can be set before starting the VDM only.

DOS_UMB

If this parameter is set on (the default is off), then DOS will manage Upper Memory Blocks in the twilight zone and make them available both to applications and to the DEVICEHIGH= and LOADHIGH statements. Occasionally applications want to directly access these areas and will not run when DOS_UMB is set ON. This parameter can be set before starting the VDM only.

DOS_VERSION

The DOS_VERSION setting specifies a list of programs and the version number which the system should report when queried by these programs. This allows applications which expect a particular version of DOS to run, since the DOS emulation simply responds with the appropriate version number regardless of what the system actually is. This parameter can be set at any time before the problem application is run.

DPMI_DOS_API

Under OS/2 2.0, an extension to the DPMI specification, known as DOS API services has been implemented. This frees the application from having to set up buffers before issuing some DOS and BIOS API calls, instead allowing protected mode applications to use protected mode buffers, referenced by protected mode selectors. The translation services perform any necessary copying of the buffers.

Applications which are aware of these extensions can turn them on or off if this parameter is set to AUTO. Alternatively, it can be set to ON or OFF. This parameter can be set before starting the VDM only.

DPMI_MEMORY_LIMIT

Up to 512 MB of DPMI memory is available in each VDM, although the default is 3 MB. This parameter can be set before starting the VDM only.

EMS_FRAME_LOCATION

Normally, the EMS page frame is located in a 64K block in the VDM's twilight zone. However, this will occasionally clash with a hardware device which the application needs to use. If this occurs, the first strategy should be to use the EMS_HIGH_OS_MAP_REGION to reduce the extra address region to zero. If this does not fix the problem, then the page frame can be relocated to one of the listed addresses.

A useful technique is to reduce the size of the DOS_RMSIZE setting and then locate the page frame below 640 KB where it is almost guaranteed that no devices can be present to cause clashes.

This parameter can be set before starting the VDM only.

EMS_HIGH_OS_MAP_REGION

This setting is used to increase the size of the EMS buffers over the default page frame. It should be used in conjunction with the MEM_EXCLUDE_REGIONS and MEM_INCLUDE_REGIONS settings.

This parameter can be set before starting the VDM only.

EMS_LOW_OS_MAP_REGION

Some multitasking environments, e.g. Windows and Desqview, can use back-filled memory below 640 KB. This setting enables this feature and specifies how much memory is to be regarded as back-filled. The default is 384 KB.

This parameter can be set before starting the VDM only.

EMS_MEMORY_LIMIT

This setting controls the amount of expanded memory available in a VDM. Setting this to zero will disable EMS in that VDM; setting it to a huge value — e.g. 32 MB — simply wastes memory and causes swapping.

This parameter can be set before starting the VDM only.

HW_NOSOUND

This setting enables and disables sound output from programs. Sometimes sound is degraded by the systems timeslicing, at other times multiple programs compete for speaker access and all but

THE OS/2 DOS ENVIRONMENT PART III

one should be disabled. This parameter can be set at any time.

HW_ROM_TO_RAM

This parameter causes the system to copy the machine's ROM BIOS into RAM for faster access. This is the same feature as the ROM Shadowing provided by some system BIOSes.

This parameter can be set before starting the VDM only.

HW_TIMER

When this setting is enabled, applications can have direct access to the 8253 counter-timer circuit ports. With it disabled, access to the 8253 is virtualized, causing a delay between the timer being read and its value returned to the VDM so that readings are inaccurate. This means that some applications — for example, games — may run

unacceptably slowly or may not run at all unless allowed direct access to the timer.

Although this setting can be changed at any time, it may not work correctly if an application programs the timer while it is virtualized and then reads it with direct hardware access. Nonetheless, it can be useful to experiment with this setting and observe the changes in performance.

IDLE_SECONDS

The operating system watches for programs performing rapid polling of the keyboard, and when it sees this, reduces the priority of the application on the assumption that it is idling. However, some programs may pause briefly to allow user input and then resume normal operation, and if the priority has been dropped, performance will be degraded. To override this, this setting disables the

IDLE_SENSITIVITY function for a number of seconds after useful work is detected.

If a program appears to run slowly when waiting for user input, this value can be increased.

IDLE_SENSITIVITY

The idle sensitivity is the percentage of the maximum possible polling rate above which an application is considered to be idling. The default is 75% — if an application polls more frequently than this, then its priority is dropped. However, the application may be doing useful work while polling the keyboard frequently, and a priority drop would slow it down unacceptably. If this is the case, this setting should be increased. Setting it to 100% turns off idle detection completely.

KBD_ALTHOME_BYPASS

In the default setting (Off), the Alt-Home keyboard combination will switch a DOS session between full-screen and windowed modes. With it set to ON, this keystroke sequence is ignored, and the user must use Ctrl-Esc to switch to the session manager and then use the DOS sessions context menu to switch to windowed mode. This parameter can be set at any time.

KBD_BUFFER_EXTEND

This parameter increases the size of the keyboard type-ahead buffer to the level available in OS/2 VIO windows. A few applications which read raw input will not benefit, but few will be adversely affected, so the default setting is ON. This parameter can be set at any time.

KBD_CTRL_BYPASS

This parameter is used to disable either of the Ctrl-Esc and Alt-Esc hot-key combinations which switch screen groups or activate the Window List, respectively. This allows the DOS application to use these key combinations for its own purposes.

KBD_RATE_LOCK

With this setting turned ON (the default is OFF) applications are prevented from altering the system keyboard repeat rate.

BENCHTECH

for OS/2

A Complete Performance Measurement Package for the OS/2 2.0 Operating System

EASY TO USE:

- Presentation Manager and WorkPlace Shell based
- Easy access notebook style interface
- Full on-line help, with hyper-text links
- Separate on-line documentation
- Graphical Installation

BENCHMARKING POWER:

- 32 bit 80386 Code
- 25 benchmark tests in all
- Disk drive and diskette drive tests
- Logging Facility to save and/or print results
- Computational benchmarks for integer and floating point math
- Powerful Macro Feature to facilitate test suites, using IBM Rexx
- 2 high-level benchmarks designed for use in comparing systems
- 8 video benchmarks that use the OS/2 Graphics Programming Interface (GPI)

BENCHMARKS THAT MATTER:

- Designed to access resources and OS/2 system code like OS/2 applications
- OS/2 Application Tests (requires Excel for OS/2 and/or DeScribe)
- Specify your own OS/2, DOS, or even Windows-OS/2 tests
- Examine the effectiveness of memory cache

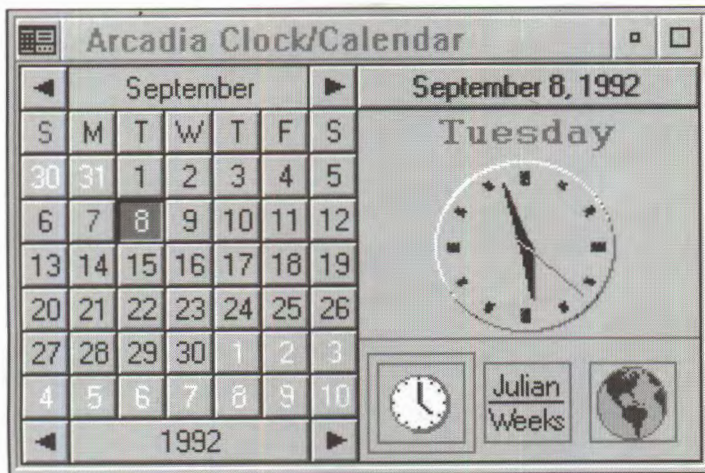
Available for a limited time for just \$79.95.
To order call
(800) 598-1718 in the continental United States or
(303) 241-1718 outside the continental United States
or FAX
(303) 243-4516
Shipping and handling charges extra. Visa/Mastercard and most P.O.'s welcome.

Synetik SYSTEMS

2210 Highway 6 & 50, Suite 204
Grand Junction, CO 81505

Arcadia Workplace Companion

An OS/2 Personal Information Management System



Clock/Calendar

- Analog/Digital Clock
- Julian/Week/Holidays
- International Times
- Custom Alarms
- Daily Notes
- Full Year Calendar

Workplace Companion Modules:

- Clock/Calendar
- Appointment Book
- Telephone/Address Book
- Contact Management/To Do List

Special

**Introductory Offer:
\$29.95**

Clock/Calendar Module

The Arcadia Workplace Companion is designed to be an integrated part of OS/2 2.0 Workplace Shell environment. It is a set of convenient and indispensable desktop utilities for every OS/2 2.0 user. The Companion products are designed specifically for OS/2 2.0 from the ground up. The user interface follows the Common User Access 1991 (CUA '91) guidelines and the conventions adopted by the OS/2 2.0 Workplace Shell. More importantly, the Arcadia Workplace Companion provides the functions you need in an ingenious

layout. Graphical buttons and value sets represent all of the choices that are available. It is so easy to use, the on-line documentation is all you will ever need.

For a limited time, the *Clock/Calendar* module of the Arcadia Workplace Companion is available at the low promotional price of \$29.95. Call Arcadia Technologies now for information regarding trade-in credit on future modules of the Workplace Companion.



**Arcadia
Technologies
Inc.**

**Tel: (818) 446-6945
Fax: (818) 447-4212**

OS/2, CUA, and IBM are registered trademarks of International Business Machines Corporation. Arcadia Workplace Companion is a trademark of Arcadia Technologies, Incorporated.

THE OS/2 DOS ENVIRONMENT PART III

MEM_EXCLUDE_REGIONS

This setting is used to specify ranges of addresses in the twilight zone which must be excluded from use as EMS pages or XMS UMBs because they are used by adapter cards such as Super VGAs or network cards. This parameter can be set before starting the VDM only.

MEM_INCLUDE_REGIONS

This setting is used to force inclusion of memory ranges in the twilight zone, even though the system might suspect that they could be used by hardware adapters. If the user knows that a DOS VDM session will not use a particular hardware adapter, then its address range can be marked for inclusion, and allocated as EMS pages or EMS UMBs to improve performance. This parameter can be set before starting the VDM only.

MOUSE_EXCLUSIVE_ACCESS

Some applications, such as Microsoft Windows and Flight Simulator, perform their

own mouse management, and if their mouse sensitivity or speed threshold settings are different from those of Presentation Manager, then the application mouse pointer and the PM pointer will drift apart, so that the user cannot tell the actual position of the real mouse pointer. The user can fix this problem by turning this setting ON, and then clicking either mouse button inside the VDM window. This will turn off the PM pointer. It can be recovered by pressing Alt, Ctrl-Esc or Alt-Esc.

PRINT_TIMEOUT

This setting specifies the number of seconds between the initiation of printing and the closing of the spool file. While the default of 15 seconds is fine for screen dumps and other small print jobs, more sophisticated print jobs and complex print formatting may take longer and will therefore be split into multiple spool files, producing unacceptable print jobs (page breaks in the wrong places). The timeout can be set to any

period between 0 (no timeout) and 3600 seconds.

VIDEO_FASTPASTE

With this setting enabled (default is disabled), the Paste menu option will insert text from the clipboard at high speed. This can over-run the buffers of applications such as the Microsoft Programmers Workbench and its precursor, the M editor, and other applications which monitor the keyboard interrupts directly.

VIDEO_MODE_RESTRICTION

If the user knows that an application will always keep an EGA or VGA video card in MDA or CGA mode, there is no need for base memory to end at 640 KB - it can extend up to the base of the MDA at B0000H, giving an extra 64 KB of user memory, or even to the base of the CGA at B8000H, for an extra 96 KB of memory. Possible settings are NONE, MDA and CGA. If an application unexpectedly switches the virtual display

Find Code *Fast* • Change Code *Fast*
Learn Code *Fast* • Port Code *Fast*

SourceLink™ for OS/2

SourceLink is an OS/2 programming development tool combining the speed and power of hyper-link source code access with a fully functional editor. The edit screens are context sensitive. Click to access on-line help. Click-click to access your source code. With over 100 menu items to choose from, SourceLink is packed full of programming features.

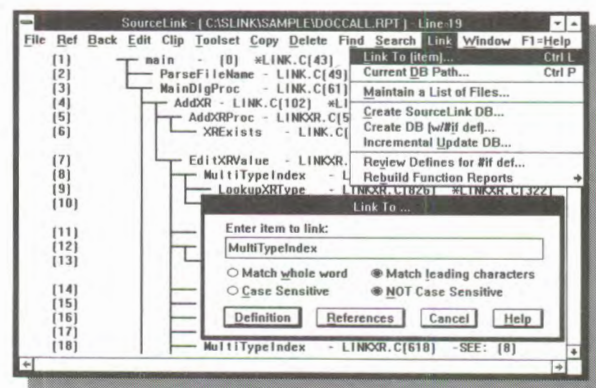
Double click on a function, symbol or other string. Let SourceLink pop up a list of occurrences. Double click on each occurrence and SourceLink will immediately display the source code. Use the integrated editor to change or create new code.

Unmatched in speed and convenience for finding and changing source code. Ideal for porting code, changing code, analyzing code, and creating programs from existing code, templates and samples.

SourceLink will save you valuable programming time and energy! SourceLink is a refreshing new experience in programming!

Features:

- HyperLink source code access.
- Fully functional editor.
- Supports 'C', C++, MASM.
- REXX macro language interface.
- Presentation Manager GUI.
- WorkFrame/2 compatible.
- Generates Function Call Trees.
- Context sensitive edit windows.
- Many file manipulation utilities.
- Find File, Delete, Copy Files/ Directories.
- Context sensitive View Help.
- Multi-File/Dir. String Search.
- An OS/2 2.0 32 bit Application.



SourceLine Software, Inc.
7770 Regents Rd #113-502
San Diego, CA 92122
(619) 587-4713

Your Satisfaction is Guaranteed

into a high-res graphics mode, this may overwrite memory now used by the application, so use with some care. This parameter can be set before starting the VDM only.

VIDEO_ONDEMAND_MEMORY

With this setting turned ON (default is OFF), the system delays allocating a video buffer for the worst-case (high-resolution graphics) video modes. This means that for many systems operating in low-memory conditions, considerable memory can be saved. However, if an application switches into a high-res graphics mode, allocation of a video buffer will be attempted. If there is insufficient memory in the system, the DOS application will be frozen and unless the user is able to reclaim memory by closing other applications, it may be unrecoverable with a risk of lost data. Use with care.

VIDEO_RETRACE_EMULATION

With this setting turned OFF (default is ON) some applications which show perfor-

mance degradation because of the way in which they poll the video retrace port will speed up. However, the drawback is that screen-switching will not be as reliable.

VIDEO_ROM_EMULATION

With this option turned ON, the system provides a 32-bit emulation of the ROM BIOS INT 10H video functions, rather than using the machines ROM BIOS. This is possible because the VIO subsystem design is basically modeled on the INT 10H BIOS functions anyway. This will provide a dramatic performance boost for applications running in a window. However, there is a possibility of incompatibility with video cards which implement enhanced functionality.

VIDEO_SWITCH_NOTIFICATION

When this setting is turned ON (default is OFF) applications are notified when they are switched to or from full-screen mode. This will cause them to redraw their

screens. However, the OS/2 screen driver will usually redraw the screen automatically in any case, so re-drawing it again simply causes performance degradation. Windows 2.x and 3.x can utilize this notification, as can applications which are written to the MS-DOS 5.0 Task Switcher API. This setting can be changed at any time.

VIDEO_WINDOW_REFRESH

This parameter sets the window update rate. Slowing it from the default of 0.1 seconds can provide more CPU time for the application, but making it very slow can render the application near-unusable. However, this setting affects only normal text-type output and does not affect screen updates based on keyboard activity or synchronous scrolling operations.

VIDEO_8514_XGA_IOTRAP

When this setting is turned ON, the system allows the application direct access to the 8514/A display adapter hardware for faster operation. This also eliminates the overhead of the 1 MB virtual video buffer. However, switching away from the application will freeze it, and the system may not be able to switch back reliably (unless the application responds to the VIDEO_SWITCH_NOTIFICATION callback above). An application run in this mode cannot be windowed. This parameter can be set before starting the VDM only.

XMS_HANDLES

This parameter specifies the number of XMS handles available for Extended Memory Blocks. It should be increased from the default of 32 only if an application uses a lot of XMS handles. This parameter can be set before starting the VDM only.

XMS_MEMORY_LIMIT

Specifies the global and per-VDM XMS memory limits. The default setting is 4 MB overall and 1 MB per VDM. This parameter can be set before starting the VDM only.

XMS_MINIMUM_HMA

This setting specifies the minimum amount of memory an application must request before being granted the HMA (since the HMA is not shareable, an application which uses only a small part of it is wasting an important resource). The default value is zero.

Oberon Terminal Emulator/2

Version 1.21

Designed and Developed by / Terminal Emulator/2 is a Product of:
 Brado Flowers / Oberon Software, 518 Blue Earth Street
 Copyright 1992 by / Mankato, MN 56001. Call 1-507-388-7001
 Oberon Software / 24 Hour a day BBS line at 1-507-388-1154

✓ XModem, XModem-1K, YModem, YModem-G, ZModem, CIS_™ B-Plus
 ✓ ANSI Terminal, VT100, IBM3101, Standard TTY
 ✓✓ Low \$65.00 Price ✓✓ Free technical support!

Oberon Software
 518 Blue Earth St.
 Mankato, MN 56001

For information call:
 1-507-388-7001
 Credit card orders call PSL at:
 1-800-242-4775

THE OS/2 DOS ENVIRONMENT PART III

This parameter can be set before starting the VDM only.

Virtual Machine Boot

Although the DOS settings above should allow almost all DOS programs to run with at most a little tweaking, there will be some that pose some problems. For example, a few programs depend on DOS data structures which are not present in the OS/2 DOS emulation. Some use undocumented functions, or features. A particular problem for the DOS emulation is programs which require a block device driver, since the DOS emulation only allows character device drivers to be loaded.

However, the VDM component of OS/2 2.0 is able to emulate a complete 8086-based PC: processor, display, keyboard, interrupt controller, DMA controller, ROM BIOS and other supporting hardware. It is therefore possible for the system to create a virtual machine which is used to boot genuine DOS in the normal fashion, by loading the boot sector of a floppy or hard disk logical drive and allowing the bootstrap process to continue. In fact, the virtual machine can boot any version of DOS, DOS clones such as DR-DOS 5.0 and 6.0, other operating systems like CP/M and Concurrent DOS or even a PS/2 Reference Diskette (though the Reference Diskette will not be able to change the configuration of the real machine, nor should you attempt to run diagnostics).

Multiple VMB sessions can be started, with multiple different operating systems running simultaneously, a real benefit for software developers testing compatibility.

A VMB is created by copying an existing DOS windowed or full-screen session, changing the Program title field in its Settings Notebook to show that it is a genuine DOS session, and then setting the DOS_STARTUP_DRIVE field to reflect the location to boot from. Possible settings for this field include:

Startup Drive Meaning A: Boot from the floppy in drive A:
C: Boot from the primary DOS partition of drive C:
C:\BOOTIMG\DOS50.VMB Boot from the specified boot image file

Typically, ones first experiments are done using the A: drive. Take an existing DOS bootable floppy and try it: a plain vanilla DOS boot floppy should come up with no trouble at all - the only things to watch out for are device drivers for expanded or extended memory, such as HIMEM.SYS and EMM386.EXE, which should be replaced with the equivalents supplied as part of OS/2 2.0. A similar MOUSE.SYS driver is supplied for mouse access.

However, if VMB will be used regularly, the user should install DOS (or multiple versions of DOS) onto partition(s) on the hard disk, using the OS/2 2.0 Boot Manager utility. DOS must be installed on a primary partition on the first fixed disk, while OS/2 can be installed on an extended partition on any drive in the machine.

Of course, this gives rise to another problem: when booting real DOS, the user wants access to extended memory, using the DOS 5.0 EMM386.EXE. But under OS/2, the user wants to use the stub device driver, C:\OS2\MDOS\EMM386.SYS, which calls OS/2s memory management services. However, there is no need to maintain two sets of CONFIG.SYS (or AUTOEXEC.BAT) files. Simply list both sets of device drivers in the CONFIG.SYS file: DOS ones first, then the OS/2 ones. When booted under OS/2, the DOS device driver cannot get access to extended memory and does not install, but the OS/2 one succeeds. Under DOS, the DOS device drivers load correctly, while the OS/2 drivers detect that they are not running under OS/2 and do not install.

Now comes another problem: how can a copy of DOS, which understands FAT file system only, access an OS/2 HPFS drive? For that matter, how can older copies of DOS understand the large drives formatted by newer versions?

OS/2 solves this problem by using a DOS/OS/2 device driver called FSFILTER.SYS. This device driver, when loaded, acts in a very similar fashion to a network re-director. DOS access to a local drive which it understands is allowed to proceed to the Virtual BIOS and access the disk through the virtual machine. However, access to HPFS and large drives is trapped and redirected through the OS/2 kernel, which will return file handles, result codes

and data just as though a network drive was being accessed. This even applies to access to genuine network drives, which is redirected to OS/2 and thence to the real network re-director.

Drive letter allocations can be mapped using the FSACCESS.EXE utility, to ensure that both DOS and OS/2 sessions see the various local and remote drives and partitions in the same way. Judicious use of the VOL and LABEL commands helps a lot here.

Another technique for allowing DOS to reside on an HPFS-formatted drive is to create a virtual machine boot (VMB) file. First, a bootable DOS disk with appropriate device drivers is created and checked by booting off drive A:. Then a VMB file is created from the floppy with a command such as

```
vmdisk a: c:\os2\mdos\dos5\dos5.vmb
```

The resultant 1.44MB file contains an image of the boot floppy; if the DOS_STARTUP_DRIVE of a DOS program reference object is set to point to it, it will boot as drive A:. As long as the file is not marked read-only (desirable in some circumstances) you can write to it, edit its CONFIG.SYS file, COPY files to it and even run CHKDSK on it!

Using this technique, I have been able to teach DOS courses running PC DOS 3.3, Compaq MS-DOS 3.31 and MS-DOS 5.0 all simultaneously. In fact, if I hadn't pointed it out, attendees would never have realized that the copies of DOS were in fact accessing an OS/2 HPFS-formatted hard disk!

A Better DOS Than DOS?

So, is it a better DOS than DOS? In recent months, I have explained and demonstrated these features of OS/2 to six different seminar audiences in the UK and Australia, and then asked their opinions. To date, the response has been unanimously in favor of OS/2. There's no way I can create a DOS 5.0 system with over 730,000 bytes of free memory. My tests indicate that OS/2 is out-performing DOS in database benchmarks by at least 40%. And under OS/2, I can allocate program 32 MB of LIM 4.0 EMS memory, 16 MB of XMS and INT15H memory, and 512 MB of DPMS memory - something that is going to be difficult with DOS!" ♦

DISCOVERING THE WORKPLACE SHELL

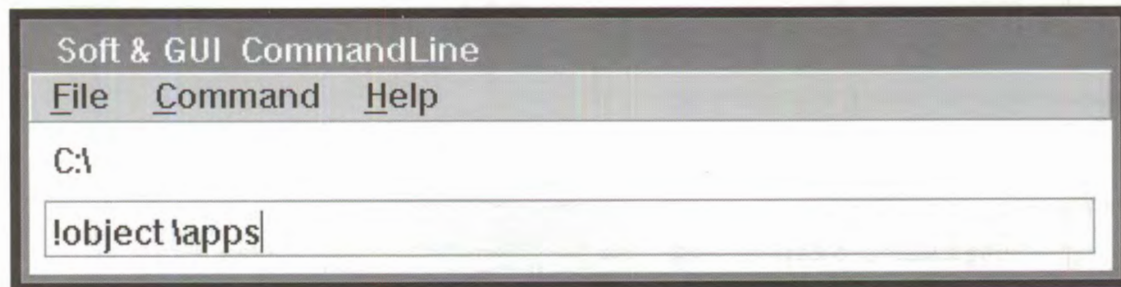
Phi-phenomenon

BRETT KOTCH

In its early years Gestalt psychology seemed revolutionary and aroused much controversy, but by mid century it had ceased to exist as an entity all to itself with many of its main lessons and factual discoveries being absorbed into the mainstream of psychology.

The founders of Gestalt psychology, Max Wertheimer, Kurt Koffka and Wolfgang Köhler argued that the nature of the parts is determined by, and is secondary to the whole. Such whole qualities have always been recognized but their explicit experimental study was not fulfilled until Gestalt psychologists began studying the phi-phenomenon. With the in-depth analyses of two stationary lights blinking in brief succession and their impression of movement, Gestalt psychology came into prominence.

Regardless of the impact that Gestalt psychology has on our life, there are still some areas of science which are just now slowly starting to learn a lesson from the Gestalt doctrine. With the introduction of the graphical user interface (GUI), and the desire to build a better GUI, system designers are eager to distinguish their newly formed interfaces from their character based predecessors.



When the discussion about user interfaces is entertained, it usually centers around two schools of thought, graphical or character-based modes. I would like to postulate another possibility, a graphical PLUS character-based mode. The combination of the two can yield a far better GUI than either one can deliver alone.

Designers must realize that as superb as a GUI might be, it will never replace the command user interface (CUI). Especially now with the proliferation of CUI's available for OS/2, (the Hamilton C shell, the Thompson Toolkit and its implementation of the Korn shell and the MKS toolkit and its implementation of the Korn shell along with all of their tools) it would be a shame to divorce the two technologies.

When I organize each month's issues into the various stages of the editorial cycle, it is much easier to use the graphical abilities of the WPS to get the job done. Links provide a vital role in allowing me to represent one article in a directory labeled by that authors name and in a directory detailing the state in which it is in the editorial cycle, and at the same time, with only one copy. By making the folder a work area, I can close and open up all of the folders that I had been using in my previous session with just one click. Although, if I want to inspect a directory which is not part of my regular agenda, it entails finding the drive icon, and profusely clicking until I get to the desired folder. The end result is a desktop full of windows.

Some developers have recognized this and have begun developing programs to bridge this gap. One is CommandLine. Instead of fondling the mouse to get the desired folder, its a hot-key away. Once you have received the CommandLine prompt, typing in "!object" followed by the directory you want sends a shadow of the folder to the desktop and opens that folder.

In the process of writing this article I stumbled upon a very powerful integration of the CUI and the WPS built right into OS/2, but that's another article. I hope that this trend towards a marriage between the GUI and CUI continues. Regardless of how advanced the WPS becomes

Zippy Object Oriented Memory™

The ONLY OODBMS for Smalltalk/V priced under \$1000 that delivers Persistent Object Storage on Disk via a Zippy B+Tree Database Retrieval Engine!



available for Smalltalk/V
(Win, PM, Mac & 286)
All Platforms \$199.95
Source Code Included!

HIERARCHICAL APPLICATIONS LIMITED

"Working to Make Smalltalk a Better Place to Live"

(512) 837-2117 or (800) SKY-PAGE Pin 5466843#

12407 North Mopac, Suite #100-266, Austin, TX 78758

INTRODUCING: dBase Read-Only File Streams for Smalltalk/V (Win & PM) \$49.95 for Smalltalk-80 Rel 4.0 \$149.95 Source Code Included!

PHI-PHENOMENON

or any of its predecessors without an interface to a CUI, we are only seeing stationary lights.

• CommandLine

Soft & Gui Inc.
224 East 21st Street
Brooklyn, New York 11229
(718) 769-8017

• The Hamilton C Shell

Hamilton Laboratories
13 Old Farm Road
Wayland, MA 01778-3177
(508) 358-5715

• The Thompson Toolkit

5616 SW Jefferson
Portland, OR 97221
(503) 224-1639

• The MKS Toolkit

35 King Street North
Waterloo, Ontario N2J 2W9
Canada
(519) 884-2251

Tennis Pros, Golf Pros And OS/2 PM Programmers Have One Thing In Common...

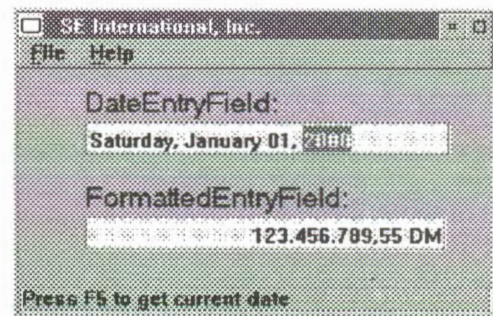
...they need proper training and tools to be successful !

The right mix makes all the difference! PM programming is easier than you think IF you use proper object-oriented techniques and high level PM window classes.

Ask for SE International's

- ✓ **Inhouse PM programming training.** Within one week you learn how to succeed in large C and PM projects and how to choose the right development tools.
- ✓ **PM Extensions.** Powerful new 16-bit and 32-bit PM window classes (**FormattedEntryFields**, **DateEntryFields**, **PrimaryWindowClass**, **Animated Bitmaps**...) allow you to focus on the application logic and not on programming of frequently required window behaviors. All our PM Extensions are Gpf™ compatible.

SE International, Inc.
621 NW 53rd Street, Boca Raton, FL 33487
Tel: (407) 241-3428 • Fax: (407) 997-8945



SES Software Engineering Services GmbH Berlin
Koenigsallee 49, D-1000 Berlin 33, Germany
Tel: +49 30 826 40 63 • Fax: +49 30 825 30 14

™ Gpf is trademark of Gpf Systems, Inc.

REVIEW

- **PrntScr**
- **Gpf**

**RON
BEAUCHEMIN
&
PAUL
DUNCANSON**

PrntScr

If you have ever had the opportunity to create a user manual for software, then you have certainly found the axiom "A picture is worth a thousand words" to be true.

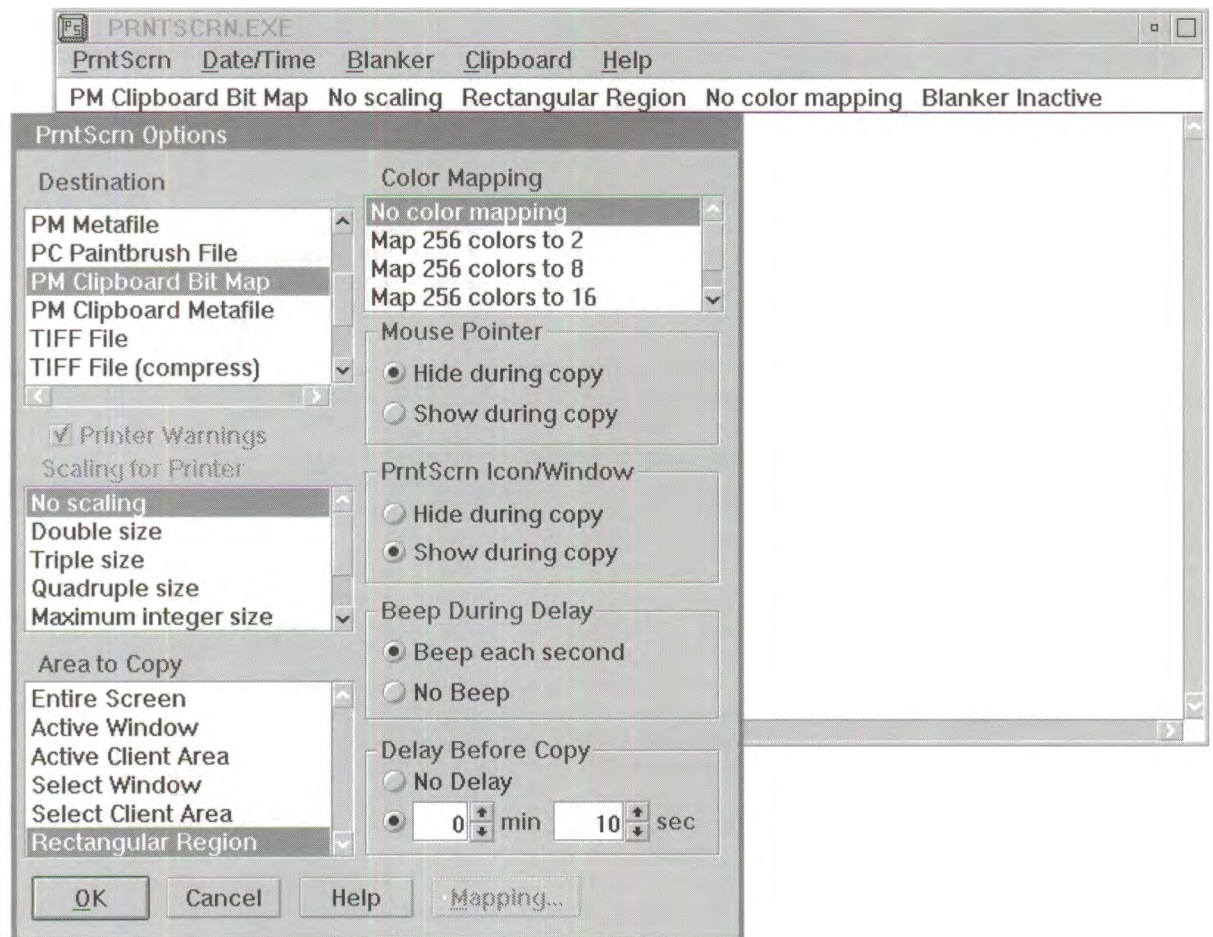
In the case of a user manual for computer products, the pictures are nearly always full or partial representations of a computer screen. Your first reaction might be that you can simply print the screen. However, the Print Screen key is an all or nothing proposition — your only choice is to get a print of the entire screen.

In many cases that is inappropriate. For example, when there is a need to focus the user's at-

tention on a dialog box or pop-up information box, a print of the entire screen might confuse the reader and your instruction would be less effective.

Fortunately there is a product that addresses the need — PrntScr by Mitnor Software. It is shipped with both the 16 bit and 32 bit versions on the same diskette.

PrntScr comes with a 52-page manual that is very well illustrated and easy to use. The manual has two installation sections which address the differences between the OS/2 1.x Presentation Manager interface and the WorkPlace Shell interface delivered with OS/2 2.0. Users who are new to OS/2 2.0 will appreciate the fact that the install



REVIEW

procedure takes them step by step through the process of launching the install program from the WorkPlace Shell.

The installation process determines which version of OS/2 is being executed on the machine, and based on that determination, either installs the 16 bit version (OS/2 1.3) or the 32 bit version (OS/2 2.0). The installation process is quick and easy.

Once installed, PrntScrn can be invoked by placing it in the startup folder or by double clicking on it's icon in the same manner as you would for any other application. Once this is done, PrntScrn is ready for use. The user can change the various options of PrntScrn at any time by selecting it from the OS/2 task list or by double-clicking on the minimized icon. PrntScrn provides the capability to capture the entire screen, the current active window, the client area of the current active window, a selected window, the client area of a selected window or any specified rectangular area.

PrntScrn provides a color mapping facility that allows you to map the captured screen area to a sixteen color image, an eight color image, a non-dithered black and white image, or a dithered black and white image. In addition to the color mapping, a printer scaling feature is provided. The scaling affects only the printed image which means that images saved to disk are unscaled. This is important because it allows you to scale up an image to multiple sizes without having to capture it again. The scaling feature uses integer scaling. This means that the scaling must be in whole-number multiples of the original size. The scalings provided are double size, triple size, quadruple size and maximum integer size. Maximum integer size is actually an automatic print-to-fit function that selects the largest integer size multiple that will allow the captured image to be printed on the current size form.

PrntScrn can also save the captured area to disk in one of ten formats. The formats are WordPerfect graphics, Presentation Manager 1.2 bitmap, Presentation Manager 2.0 bitmap, Presentation Manager Metafile, PC PaintBrush, TIFF, TIFF compressed, TIFF Grayscale, CompuServe GIFF and Windows bitmap.

The PrntScrn capture function is initiated by pressing the Print Screen key. The disposition of the captured area is determined by

the current parameter settings within PrntScrn.

PrntScrn as has an import facility that allows the user to import Presentation Manager 1.2 and 2.0 bitmap files, Windows bitmap files, Presentation Manager Metafiles and text files into the clipboard. Once a file has been imported, it may be pasted to the clipboard and saved in any of the ten file formats mentioned earlier. I tested this feature by importing a text file, the OS/2 2.0 logo bitmap and the Windows bitmap FROG.BMP that was distributed by Microsoft on the Windows 3.1 Developer Kit CDROM. I encountered a small problem importing the Windows bitmap from the CDROM. PrntScrn encountered an access error. I reported the problem to Mitnor and they are fixing it. Once I used the OS/2 copy command to copy the Windows bitmap from the CDROM to the hard disk, everything worked fine. Although not specifically mentioned in the documentation, PrntScrn can be used as a conversion utility to convert any of the import formats to any of the output formats.

A moveable date/time window and a screen blanker are built into the product. These features are user selectable at any time. The screen blanker will blank Presentation Manager and OS/2 full-screen or OS/2 windowed sessions. The blanker has no effect on full-screen VDM applications.

The authors thought about customer support when they designed the product. They built in a debugging capability that can easily be invoked by the user. This feature can be used to help Mitnor Software technical support personnel quickly resolve user problems should they occur.

I have run PrntScrn under OS/2 2.0 both on a stand-alone PC and on a PC connected to a Novell 3.11 token-ring network. The network workstation uses the OS/2 2.0 Novell Requester and utilizes a LaserJet printer attached to the network. The product worked fine in both environments. The product can also be loaded on the network server and run as a server application, but I did not test it in that mode.

There were no noticeable performance impacts while running PrntScrn along with other applications. Utilizing CPU Monitor from Bon Ami Software (which will be reviewed in a later article) I was able to deter-

mine that PrntScrn utilized 7% of the CPU on a Northgate 386 33 MHz machine. The 7% figure is a worst-case scenario. The 7% level was reached while PrntScrn was writing the screen image to disk. The file was written to disk very quickly and the CPU utilization for PrntScrn dropped to zero. PrntScrn provides the user with an audible beep to signify that the capture operation is complete. As a final test, I invoked PrntScrn while a CD-ROM audio track was playing in a Virtual Boot VDM. As expected with OS/2 2.0 there was no impact whatsoever on the CD-ROM application.

I would recommend this product to anyone who prepares documentation, user manuals or procedures for Windows or OS/2 applications running under OS/2. — **RB**

Editor's note

For those of you who would like to see a sample of this product before you consider purchasing it, OS/2 Monthly has used PrntScrn exclusively since the last issue.

PrntScrn is produced by Mitnor Software and is priced at \$115. Mitnor Software can be contacted at 918-357-1628. The FAX number is 918-585-3303.

Gpf

Case tools have long been a good idea whose reality has never quite lived up to the promise. They have always taken a long time to learn and have had limited utility. The 4GL's I've tried always seem to run out of gas just when I need them the most, forcing me to revert back to C and just about start over! They're OK for prototyping but lack the capability of producing a "World Class Application."

Much (for some applications, even MOST) of the work of writing a PM (Presentation Manager) Application or any GUI (Graphical User Interface) is producing the GUI itself. I've seen OS/2 PM applications where the percentage of code to create the GUI portion of the application varied from 5% to 95% of the total code. Add to this the complexity of creating the hypertext help data (in an IPF file format) and making Resource files to reference text strings, icons, and other resources. Also add

♦

DO YOU REALLY
THINK WE'D LET YOU
GO TO MANAGEMENT
TO GET MONEY FOR
AN OBJECT DBMS
WITHOUT GIVING YOU
AMMUNITION
ABOUT WHY IT
MAKES BRILLIANT
FINANCIAL SENSE?

♦

Tell your management to forget all this nonsense about how the production reality of object database management systems is still years away. Tell them that ONTOS already has some 400 customers worldwide.

Let them know that some of the most widely respected leaders in communications, manufacturing, financial services, health care, and government are experiencing dramatic productivity gains and increased competitiveness—this very day—thanks to the inherent benefits of the ONTOS DB.

For a copy of our brochure, *The Impact of ONTOS*, phone 1-800-388-7110, Ext. 207. And when management asks you what the ONTOS Object DBMS is going to do for the enterprise, tell them...well, tell them to fire up their calculators.



OS/2 2.0 printing as simple as...

1.

Font Cards
for IBM/Lexmark
LaserPrinters

2.

Font Cartridges
for Hewlett-Packard
Laser Printers

3.

Soft Fonts
in tape or diskette for
various laser printers

512.490.2240

Architext has a library of over 10,000 fonts, also available are custom fonts and font modifications . Font formats include Bitmap, PostScript, TrueType, and Intellifont. ATM compatible. With Architext you have the freedom to create your font card or cartridge suited to your needs. Choose from fonts, bar codes, logos, and/or signatures. A Graphic Artist will work with you in choosing the fonts and graphics to fulfill your printing specifications. *Let us help you make printing easy and affordable.*



121 Interpark Blvd Ste 208 San Antonio, TX 78216-1808 512.490.2240 Fax 512.490.2242

that most business applications must reference a data base and that means SQL (Structured Query Language) mixed with C in a SQC file. A good OS/2 PM programmer is not going to be produced overnight. Learning the basics will take six months or more on one's own.

Developing a 32 bit OS/2 PM app means using the IBM C Set/2 and the IBM Workset. As good as the IBM Workset is, and even with 8.5 megabytes of excellent sample code, not everything is covered. The next hurdle facing OS/2 is to produce enough quality applications to show the superiority of applications that fully utilize the features of the operating system. But, with a steep learning curve and a real shortage of qualified OS/2 v2.0 PM programmers, how will this demand be met?

The answer is Gpf (GUI Programming Facility) from Gpf System, Inc. My first close-up look at Gpf was in May of 1991 at the first of the IBM OS/2 v2.0 Technical Seminars in Texas. It looked really promising, but with a \$3600 price tag and only a 16 bit version for OS/2 1.x, I decided to wait. As I have said, I've bought things that look good in the past only to be disappointed and be out the money too. Well, Gpf has lowered the price to \$995.00, added features, and recently released a OS/2 v2.0 32 bit version that not only does it all, but does it all amazingly well. Anyone doing serious OS/2 v2.0 PM simply must buy this product. It will save it's entire cost in the first week!

The value of Gpf is 3 fold:

1. As a programmer training tool.
2. As a rapid prototyping tool.
3. As a GUI visual programming tool or GUI screen painter to automate the production of C, IFP, Resource, and SQC files.

The first step in using Gpf in any of the three ways is to invoke Gpf and load an existing project in development or to invoke the menu to define a new project. You simply give the project a name, choose a copyright message and compiler, and indicate whether or not you will be using the OS/2 Database Manager. Then, you start opening windows and dialog boxes and placing controls in them. The objects menu item of the Gpf editor lets you select and define the characteristics of Message

Box, Presentation, Icon/Bitmap/Pointer, User Function, and User Control objects. A presentation object, for instance, would let you define a font style, size, foreground and background color as a building block. The create menu item of the Gpf editor lets you select a Window, Dialog Box, Menu, and all the standard as well as the custom controls. If you select a "Slider" control, you can place it in a window where you want and size it. A full screen options window then opens up with check boxes, radio buttons, edit fields, etc. that enumerate all the possible variations and options available in PM for that control. You can also go to an "Actions" window to select many standard actions to be associated with that control. For various types of windows and control, these actions can be as simple as setting the focus, or as complex as querying the database and placing items from it in a MLE (Multi Line Edit field). An even more complex action require you to write your own C code to implement the algorithm.

As a programmer training tool, it produces a quality, well-commented C code, as well as an SQC, IPF, definition, resource, and C header files. The OS/2 PM paradigm is so rich that even an experienced PM programmer may forget how to implement some features. It may be quicker to simply draw the GUI features with Gpf and let Gpf generate the source files. New programmers can get tremendous training benefits by examining and studying the Gpf-produced source file.

As a rapid prototyping tool, Gpf has an "Animate" mode and a "Test" mode for checking the look and feel of the GUI interface you've drawn before any source code is generated. The Test mode is the simpler of the two, allowing only simple selection of value sets, containers or push-button, etc. The Animate mode allows such complex actions as the display of list boxes that actually contain data from your SQL database.

Once you have your applications PM interface prototyped approximately as you want it, you can let Gpf generate the source files. Here, you are using Gpf as a visual programming tool. This is as simple as selecting GENERATE from the TOOLKIT menu bar item of Gpf. Gpf generates all the different files required to generate the GUI portions of your program. It also generates

the shell code for the algorithm portions of the program. You, of course, must generate the source code to actually implement the algorithm portion.

Gpf allows three basic techniques for merging your own code to the Gpf produced code.

1. Directly modify the Gpf generated source code;
2. Create a user control object;
3. Create a user function object.

The problem with directly modifying the Gpf-generated source code is that if you decide to revise any aspect of the GUI interface using Gpf, all your modifications to the code will be lost when you generate the source files again.

You can create you own controls and use them as standard program-building blocks. If you have a need for a circular gauge type control, you can create one and use it as a standard building block just like the sliders and push-button controls.

The creation of user function objects is the best way to merge the algorithm portions of your code, so that you can revise the GUI with Gpf and still not lose your additions. By selecting USER FUNCTION OBJECT from the OBJECTS entry on the Gpf menu bar, you can add a new user function object. You give your new user function object a name and add up to four k of C language source code to the resulting edit box for inclusion into the Gpf-generated code. Generally, you will add only one line — a function call to a function that will encapsulate that portion of your algorithm code. Gpf knows to look for a header (.h) file with the same base name as your application for the prototype of the function. The function code itself can be compiled and placed in a library or DLL (Dynamic Link Library).

As a final step, you can invoke your PM compiler to compile and link your new program. Then, execute the resulting EXE file directly or via your debugger to debug the algorithm portion of your code. It is amazing how fast you can produce a PM application with Gpf. At the end of only a two-hour Gpf session, I've generated applications in which the C source file alone was over 225k. How long would it take you to write 225k of C source? As much as the C

REVIEW

source code generation time is cut, the debug time is cut even more. Since the GUI portion and the shell code for the algorithm are produced by Gpf bug-free, you can concentrate on the bugs you introduced in the algorithm portion you added. Unlike many 4GL's, there are no per-copy royalties on programs you produce with Gpf.

As you can tell, I am very impressed with this product and plan to make it part of my standard toolkit. There are only a couple of minor negatives. The worst is that the tutorial needs some major revisions. It took me several tries to get through it. To make sure it was not just due to my own error, I asked three other OS/2 PM programmers to try the tutorial. Two of the three required help from me get through it. When I first tried the tutorial I had to call the Gpf Systems help line. The good news is that the help line person was extremely knowledgeable and quickly got me back on track. Once you have completed the three tutorial exercises, you'll be well on your way to being competent in using Gpf. Even if you stop to analyze what you have done, and take time out for a phone call or two to the help line, you can do this in less than a day. The second negative is that a better control copy needs to be incorporated into the product. For instance, I created a text value set of the months of the year in two rows of 6 columns each. In the same dialog box, I needed to repeat this value set to get a different month. While I could copy the value set as a text value set with two rows of 6 columns each, I had to enter the names of the months again via the Gpf menus. While there are ways around this, including a utility on the Gpf BBS, a better copy needs to be integrated into the base system. But, even with a couple of minor negatives, it's hard not to love a tool that lets you do in two days, what used to take a month! Gpf is a must-have for any serious OS/2 PM developer. — PD

A free demo version of Gpf is available upon request from the company. Gpf, can be ordered from Gpf Systems Inc., P.O. Box 414, 30 Falls, Rd., Moodus, CT 06469. Phone (203) 873-3300 or (800) 831-0017. Fax is (203) 873-3302.

GammaTech

Utilities for OS/2

Get the Best Performance and Reliability From Your Mission Critical OS/2 Work Station

The GammaTech Utilities are designed to enhance and complement your OS/2 system. Several utility functions are implemented in a Presentation Manager shell while others are provided in a command line mode for quick and easy access. These utilities provide the ability to perform the following maintenance and recovery operations easily without extensive technical knowledge:

- Easy Graphical Installation
- SAA CUA Compliant PM Interface
- Optimize HPFS Volumes
- MS LAN Manager Support
- Recover Erased Files
- Obliterate Sensitive Data
- Mass Erase Selected Files
- Display Volume Information
- Alter File Attributes
- Add Comments to Files
- Alter File Time Stamps
- Locate Files
- Edit Disk Sectors
- Test Disk Media
- Display File Fragmentation
- Display Directory Information
- Supports HPFS386
- Supports Long File Names
- Supports Extended Attributes

The Best Time to Acquire Recovery Software is Before you Need it.

For OS/2 version 2.0 or 1.3. List \$125.00

VISA, MasterCard, COD, Purchase Orders
Federal Express Shipping for Fast Delivery

Post Office Box 70 Edmond, OK 73083
Phone (405) 359-1219 FAX (405) 359-7391

ON THE EVE OF INSTALLATION

BEGINNER'S SLOPE

Welcome to the Beginner's Slope! Since this is a column for OS/2 v2.0 novices and newcomers, a logical topic for the first column is how to prepare for installation. With this, as with future topics, there won't be much discussion of the bits and bytes that make OS/2 tick; the excellent columnists in other sections of this magazine can provide that. Here we'll take a more pragmatic approach to utilizing OS/2 quickly and enjoyably, and just a small amount of DOS knowledge on the part of the reader will be assumed.

The subject of this article was not chosen because installation of OS/2 is difficult. Contrary to what you may have heard, it's actually quite easy, takes about 50 minutes and usually goes without a hitch. But there are many ways in which to install OS/2, so this article will attempt to guide you through the decisions to be made. And because installing a new operating system is a big step that involves some risk no matter how stable and friendly the operating system is, this article will remind you to take a few sensible precautions as well.

Many of the following steps are recommended not just before installing a new OS, but whenever making a major change in your system's configuration, and some should be performed on a regular basis. The steps are: backup all files, make certain your hardware is appropriate for OS/2, decide on which final configuration is desired, create a bootable DOS floppy if one is not already available and, if you will be changing the hard drive partitions, practice repartitioning the hard drive under DOS using the FDISK command. If the machine is or will be attached to a LAN, make a record of all network addresses in case they are needed after installation, and be certain the proper LAN requester is available for the PC to access it under OS/2. In a similar vein, record all frequently used telecommunications information.

The most important of the steps above is backup. Backup is listed even before evaluation of hardware because it never hurts to do an extra backup. Perhaps a warning to backup doesn't belong in a column aimed at beginners, because it often seems that only experienced computer professionals forget to backup (!). But seriously, to backup frequently is sound computing practice and this includes taking backups of all the original diskettes that come with your various software packages.

If OS/2 is being installed "in place", that is to say, over DOS or on top of an older version of OS/2, then the main purpose of back-

ups is to restore the system to its original state if a problem arises during installation. However, if OS/2 is going onto a recently repartitioned or otherwise empty hard drive, then backups are frequently used to place previously existing applications and data back on the hard drive after OS/2 has been installed. As much as possible, try to avoid using backups to place files back on the disk, especially if data compression packages such as Fastback or Central Point Backup are being used. Reinstalling applications from the original diskettes is preferable to using backed up data for two reasons: first, a package that is installed on one drive often will not work when restored to a different drive, and second, if DOS backup/compression packages are used, there's a good chance the software will not be compatible with OS/2 v2.0. Even if your system has a large hard drive, has many files and the original disks for the applications are not available, it's best to forget about using any of the commercially marketed DOS backup packages.

If you're experienced with PCs and are not daunted by the challenge, you can experiment with Fastback or one of the other packages, but don't stake any vital data on the outcome. In any case, there are easier ways to save your data for use later under OS/2: probably the best approach is to use the BACKUP command that comes with DOS. This won't provide the options, flexibility and high compression ratios of some of the "name" packages, but there's no concern about compatibility. The complementary command to be used under OS/2 is RESTORE, and it will restore files even to HPFS format (more on HPFS later). Don't waste time looking for it in a DOS window however; RESTORE is requested from an OS/2 command prompt.

An alternative approach, if you're fortunate enough to be attached to a LAN, is to reserve some space on the server to temporarily hold data during repartitioning and installation. If there's enough room, no compression will be necessary and the simple command XCOPY will suffice. If the server is not quite that generous, then use BACKUP. Even the utility PKZIP might provide all the compression that's needed to backup data to the LAN server. On the OS/2 side PKZIP2 and PKUNZIP2 usually come installed with the system (that's right; "usually"). When restoring is done and everything is working, remember to delete the data from the server so that other people can use the space while installing OS/2. Of course, none of this applies if the hard drive will not be repartitioned or reformatted.



Bill Zinsmeyer

ON THE EVE OF INSTALLATION

Now that all the files have been carefully backed up, the hardware can be evaluated. The requirements of OS/2 are well known: at least 4Mb of RAM, a 386SX or better processor, about 40Mb on the hard drive and a mouse or other pointing device. It's also fairly well known that running OS/2 on a system with just these minimum requirements is not much fun. A configuration more likely to provide enjoyable performance would have 8Mb of RAM, a 386DX-25 processor and 80Mb on the hard drive. When estimating available space, keep in mind the possibility of deleting DOS, any memory manager and Windows (although not everyone will feel ready to do this). If you are installing Extended Services or LAN Services as well, then additional space should be kept available. If you are considering an improvement to your system in preparation for OS/2, note that as a general rule additional memory will be a greater boon to OS/2 performance than a faster CPU. After the CPU, consider improving disk speed and disk space for better performance.

Assessing the hardware

When assessing the hardware, remember also that OS/2 is one of the greatest tests of components ever invented. This means, for example, that if the CPU is an early version containing a mask error, the first time you find out about the problem may be when you start to use OS/2 (just as in software, bugs occur in CPUs). Similarly, if the RAM consists of inexpensive chips from a disreputable manufacturer or if the printer cable does not have all of the connections functioning that it should, OS/2 might be the first to detect that something's amiss. Sometimes DOS and Windows will not reveal these problems, thus giving the illusion that all is well. OS/2 is probably the first software you'll own that will utilize the components of your system to their maximum potential, so, unfortunately, it might also be the bearer of bad tidings.

Attention must also be paid to video adapters and SCSI drives. Certain video cards such as Tseng-based cards do not yet work well with OS/2. With some chips (the "S3" chip, it has been said) erratic behavior can result during mode-switching. Mode-switching is a reference to the switch into Super VGA mode for Windows under OS/2

and the subsequent switch back to VGA for OS/2. This is why a few critics claim OS/2 does not provide "seamless" Windows — because there are still some difficulties with SVGA. Just be patient and live with VGA a little longer; these problems are being solved quite rapidly, so the critics will soon have to look for some other minor point to harp on. Check with IBM or your video card manufacturer if you are concerned.

Other peripheral devices

Other peripheral devices that are somewhat out of the ordinary, such as scanners, tape drives and CD-ROM drives, can pose a bit of a challenge under OS/2, although many work with no trouble. Certainly caution must be exercised when planning to use a SCSI drive with OS/2, because SCSI device drivers (the software necessary for the device to interface with the operating system) are not plentiful at this point. Availability is increasing every day, however, so check with IBM or with the manufacturer. As for other peripherals, it's important to note that if no device driver has been written for OS/2 v2.0, it does not mean that the device cannot function under OS/2. In almost every case, the existing device driver can be defined to any or all VDMs, so the scanner or other device is available within OS/2 through DOS sessions at least. (VDM stands for Virtual DOS Machine, or a DOS box to us novices.)

The more unconventional the peripherals, the more likely you'll want to keep native DOS available in your final system configuration. Even an innocuous piece of software such as the educational games produced by Walt Disney can be reason enough to keep standard DOS in some form. Because the Disney software "talks" through a speaker attached to the printer port and because OS/2 takes much firmer control of printer output than DOS or Windows (and also because the interrupt rate for OS/2-DOS is less than native DOS) the sound quality is unacceptable under OS/2. This is not a shortcoming of OS/2: sending sound through the parallel port is a little strange anyway. Also, it's great that OS/2 has a sophisticated print spooling feature and, besides, the software in question has poor sound quality under Windows as well. Believe it or not, IBM, among all their other tasks, is working on getting the

Disney software to sound clear under OS/2. To summarize then, there are a few valid reasons why the version of DOS provided within OS/2 might not be sufficient, such as having unusual peripheral devices or not wanting to explain interrupt rates to an unhappy 3-year-old.

A purely OS/2 setup is recommended if it's practical, but if not, then the decision must be made which approach to use: Dual Boot or Multiple Boot. Dual Boot allows DOS and OS/2 to coexist in the C partition. Only one system may be active at any one time. Dual Boot, if installed over the existing DOS, has the advantage of avoiding all the time-consuming tasks involved with repartitioning and reinstalling a lot of software. A complete backup is still recommended, of course. Installing Dual Boot without any reformat and the associated replacement of data is only practical if the existing C partition is large enough.

Multiple Boot was originally intended to permit the installation of more than two operating systems (AIX could be the third, for example) but it works very well with just DOS and OS/2. Some people feel more comfortable keeping the two systems distinctly separate. Multiple Boot requires the installation of Boot Manager, which occupies 1Mb on the hard drive, and it takes advantage of the fact that OS/2 v2.0, unlike DOS, does not have to reside in the primary partition; it runs just as well from an extended partition. Here is a typical arrangement for Multiple Boot on a 120Mb hard drive: 1Mb for Boot Manager (assigned no letter), 4Mb for DOS (drive C), 65Mb for OS/2 with HPFS (drive D), and the remaining 50Mb formatted as FAT for DOS and OS/2 to share (not simultaneously) in drive E.

Other considerations

Although Dual Boot, Multiple Boot, Boot Manager and several sample configurations are described quite well in the Installation Manual, a couple of further points should be made. One important point: if new partitions are created using the FDISK feature of OS/2, DOS FDISK will be unable to remove those partitions. This only becomes an issue if installation has to be halted and abandoned for some reason, in which case the OS/2 FDISK function available on the first two or three installation disks must be

Relish[®]

Time Management Solutions for Your Desktop and Workgroup

Relish Provides:

Superb Reliability

Extensive Versatility

Unprecedented Integrity

Cutting-Edge Technology

Automatic reminders without even running Relish

Convenient and intuitive scheduling and revision

Immediate and automatic saving and refreshing

A true PM solution based on client-server architecture

Relish for personal calendaring. Relish Net for group scheduling.

Multi-tasking/multi-threaded design. Both available in 16-bit and new 32-bit Workplace Shell flavors.

Relish brings an intuitive edge to state-of-the-art time and information management. You make "notes" to schedule and track various events and categorize information. Relish does the rest. Scheduling is realistic: precisely when you want, for any time and duration, including double booking as necessary. And, Relish is easy to use. A host of time-saving features enhance the CUA compliant design.

Relish - \$149. Relish 32-Bit - \$189.

Relish Net integrates group calendaring features with personal time management. View your own calendar, someone else's, or several together. Search for free time. Schedule personal, group, and resource commitments. Optimize scheduling coordination without compromising security, personal privacy, or control. Relish Net is a server-based application (not simply a program using shared files), providing the LAN-based workgroup with scheduling and notification in real time.

Relish Net 5 user increments - \$595. Call for pricing on Relish Net 32-Bit.

Sundial Systems Corporation (310) 596 • 5121

909 Electric Avenue, Suite 204, Seal Beach, CA 90740 USA



Credit cards and corporate purchase orders accepted. Quantity pricing available.

• Special Offer • \$1 a Bit • Special Offer • \$1 a Bit • Special Offer •

New Relish 32-Bit - more features & Workplace Shell enhanced - just \$16 more than the \$149 16-bit price. That's 16 special bits of Relish for 16 bucks. Order direct. \$165 Relish 32-Bit offer expires Dec. 16, 1992.

Relish is a registered trademark of Sundial Systems Corporation. CUA, OS/2, Presentation Manager, and Workplace Shell are trademarks or registered trademarks of International Business Machines Corporation. © Copyright 1992 Sundial Systems Corporation. ALL RIGHTS RESERVED.

ON THE EVE OF INSTALLATION

invoked again to remove the partitions. Also, it is recommended that one have at least a basic understanding of primary, extended and logical drives, which are divided into C, D, E and so on.

More advanced users

For more advanced users, another variation that makes clever use of the Multiple Boot option is to reserve a partition for another copy of OS/2. This would not squander much space on the drive (only 4Mb) because it would be a greatly reduced copy of the operating system, lacking the graphical Workplace Shell. This is, in effect, an alternative to keeping a bootable OS/2 diskette on hand and, as stated, is an option more of interest to experienced users than to novices. The partition could include more files than would fit on a bootable floppy and so have greater functionality. For more information about OS/2 on a bootable diskette there are help documents available on Compuserve, on other electronic bulletin boards and from IBM.

Many hard drives will not need to be

repartitioned, but many will still have partitions of 32Mb or smaller which are holdovers from earlier versions of DOS. A common question asked when planning to install OS/2 is whether the "lower" drives C and D can be merged together and reformatted, or have their proportions relative to one another shifted, while leaving the "higher" drives E and F undisturbed. Perhaps this could be accomplished using some specially written utility to cleverly alter the file allocation tables, but for most people it is not practical. The DOS FDISK command will refuse to cooperate if asked to delete a lower drive while an extended drive still exists above it. In any case the data on C and D would be lost.

It's conceivable, with the right starting configuration, that the Multiple Boot option could be installed by merging and reformatting the higher drives, leaving DOS and other files in the lower drives undisturbed. The process would require some proficiency with PCs, however, and should only be tried on an experimental basis. It might work because Boot Manager does not need

to be installed at the beginning of the hard drive; it can also be installed at the very end.

Another major decision to be made with regard to installation is whether or not to utilize the High Performance File System ("HPFS"). The conventional DOS file system, which can still be used for OS/2, is called "FAT" for File Allocation Table. FAT was originally created as a hybrid of the file systems found in UNIX and CP/M, and although it has been greatly improved since its creation, it still retains some disadvantages from its relatively old heritage.

Main advantage of HPFS

The main advantage of HPFS is speed. The FAT system still looks for files sequentially, which can degrade access time in a large partition. HPFS uses a search algorithm and other enhancements to improve access time. FAT file names are still limited to the eight-dot-three pattern, while HPFS allows names up to 254 characters in length and permits greater use of special characters, including spaces. The FAT system is prone to file fragmentation and lost chain problems which utilities such as Norton or PC Tools are often used to remedy. HPFS greatly reduces these problems so that clean-up utilities are not necessary. HPFS maintains not just the data contents of a file but also the extended attributes associated with it (as does FAT under OS/2). Extended attributes can hold the name of the person who created the file, the creation date and time, the icon used to represent the file graphically, any executable-program-to-data-file association that's been established and other information.

Extended attributes provide many useful and interesting benefits that could be the subject of an entire article, but they won't be discussed here as they are supported by both file systems in OS/2 and therefore do not affect the HPFS vs. FAT decision. In fact, FAT under OS/2 has been improved in several respects (lazy writing and caching, to name two), but HPFS is the better choice for large hard drives. It's interesting to note, by the way, that OS/2 was designed to adapt easily to new file systems in anticipation of the day when a better one is invented.

If HPFS is clearly superior to FAT, and it is, one might wonder why consider using

Sure - you can learn OS/2 2.0 programming by the book.

But you can save time and money by attending our workshop.

It can take up to a year just to wade through all the manuals needed to learn OS/2 2.0 programming. We will have you writing a professional application immediately upon finishing one week of our intensive, hands-on training.

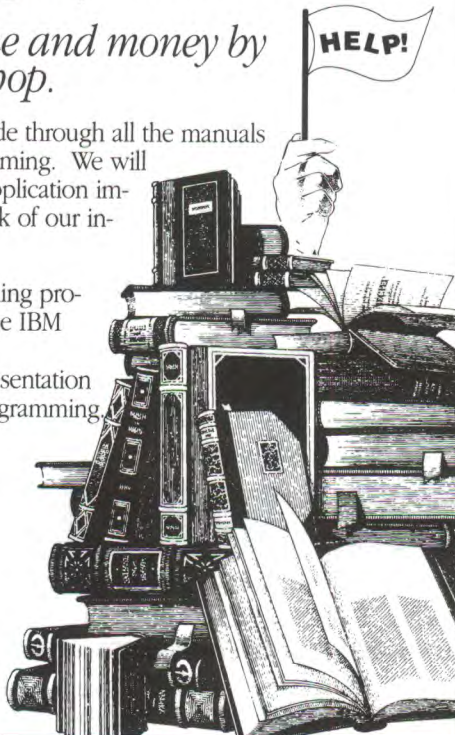
Descriptor Systems has been teaching programming classes at companies like IBM since 1983. Our catalog includes: OS/2 2.0 Kernel Programming, Presentation Manager, and Workplace Shell Programming.

Contact us for more information on these and other programming workshops.



Descriptor Systems

P.O. Box 461
Marion, IA 52302-9909
Phone: 319-362-3906
Fax: 319-362-3701



FAT at all under OS/2. HPFS does have some small disadvantages, the principal one being that if DOS is booted up independent of OS/2, all partitions formatted for HPFS will be inaccessible to DOS. The HPFS partition can seem "invisible" so that DOS, with certain configurations, will temporarily remap the higher drive assignments while it is active, thereby confusing software applications (and the user). If native DOS is used frequently it is convenient to be able to access files in all partitions. On those rare occasions when OS/2 has a problem it's often very helpful to attempt a fix from DOS, which HPFS can prevent.

The High Performance File System is not recommended on a system outfitted with less than 8Mb of RAM. HPFS ties up about 0.5Mb of RAM and also occupies slightly more disk space than FAT, at least initially. Also most DOS and Windows programs, as well as some older OS/2 programs, cannot accept long file names, although this is only an inconvenience and not a genuine reason to avoid HPFS. Long names are op-

tional, so simply stick to the eight-dot-three convention while using DOS and Windows programs, and they will work well under HPFS (and often run faster). It is commonly asked whether diskettes can be formatted to use HPFS. They cannot, and when files with long file names are copied to a FAT diskette the names will be truncated (so too when copying to a FAT partition on the hard drive). However, the long name and other extended attributes will be preserved and restored when the file is moved back to HPFS, as long as a DOS or Windows program has not updated the file and lost the extended attributes.

Bootable DOS diskette

Regardless of whether or not a major installation such as a new OS is being considered, no PC user should be without a bootable DOS diskette. To create one, place an appropriately-sized floppy in the A drive and enter the command `FORMAT A: /S`. The slash-S at the end causes the floppy to be formatted as a system disk,



which means two hidden DOS files are copied to the diskette and it is bootable. After that it's recommended that short versions of the `AUTOEXEC.BAT` and `CONFIG.SYS` files be created on the floppy and several of the more useful DOS files should be copied as well (certainly `COMMAND.COM`). Try booting from the diskette and running a program or two to be sure

Worried about testing your OS/2, networked applications? Not if you're using the Softbridge Automated Test Facility.

If you're building OS/2 (or Windows) applications, you know that software testing is half the battle in the development cycle. If your applications are networked, they're even more complicated to test.

The Softbridge Automated Test Facility (ATF) was designed for unattended, centralized testing of OS/2 and Windows applications, networked or stand-alone.

ATF is used by corporations and software vendors who want to:

- ✓ **Build higher quality software**
- ✓ **Decrease development time & costs**
- ✓ **Make the impossible possible**

Don't just take our word for it ...

Cooperative Solutions used ATF to test *Ellipse*, and "Ellipse has been practically bug free." (*Datamation* - March 15, 1992)

Reuters Information Service has used ATF for three months to test a client/server application. So far, ATF has enabled the firm to reduce the time required to perform 800 test cases from one month to one week." (*Network World* - June 22, 1992)

Bachman Information Systems chose ATF because it's 70-80% better than anything on the market." (*PC Week* - June 15, 1992)



If software quality is on your checklist, call 800-955-9190 and learn more about ATF.

Softbridge, Inc. 125 CambridgePark Drive Cambridge, MA 02140 617-576-2257

the diskette is configured properly. Note that this diskette, while handy, is not meant to substitute for a complete, official set of DOS installation diskettes.

If all other steps preparatory to installation have been performed (all files have been backed up, network addresses have been recorded for the LAN, etc.) then the bootable DOS diskette would be a good way to experiment with FDISK, assuming that the hard drive is about to be repartitioned anyway. FDISK should be one of the DOS commands available on the diskette, so at this point you could boot the system from the A drive and try FDISK from the floppy.

The installation process itself is covered quite well by the installation manual that comes with OS/2. Read the manual — it's much easier to read than most documentation from IBM — and it's so short that it barely deserves the title "manual". A couple of further points — frequently a few items are overlooked, especially if the recommended approach is taken and the software is reinstalled from the original disks. Record the telephone numbers, as well as protocols, passwords and appropriate settings from your communications programs.

For those who are frequent users of Windows, it is recommended you make copies of your favorite Windows games and applets (particularly if installing OS/2 without a DOS partition). The version of Windows that comes with OS/2, while fully functional, can appear a bit spartan due to the lack of mini-applications that Windows users are accustomed to seeing. In fact applications written for Windows are sometimes designed to utilize the mini-applications in the course of their processing. Windows games and applets work well under Win-OS/2, provided they are taken from version 3.0, not Windows 3.1.

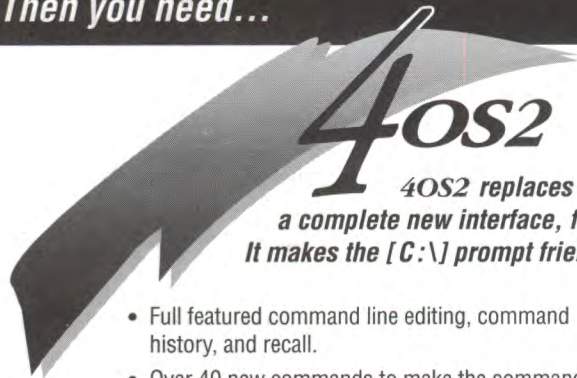
Finally, when installation is complete and OS/2 is booted up for the first time, be patient — it can take as long as 10 minutes in some cases. When the Workplace Shell is shown, the icons are visible and the disk light has at last ceased activity, DO NOT USE THE SYSTEM. *The very first action to perform is a Shutdown and subsequent reboot.* This does not mean pressing Ctrl/Alt/Delete or abruptly turning off the power. Avoid those habits with OS/2. To Shutdown, click the **right** mouse button once on an empty part of the main screen. A pop-up menu will appear and one of the

options will be Shutdown — click the **left** mouse button to activate it. Proceed with Shutdown and reboot the system. There is a remote chance of serious file corruption if this precaution is not taken. [Some call it a *bug*, others call it a *feature*, but either way, do a Shutdown before using OS/2 for the first time.]

So that's the first installment of The Beginner's Slope! If it prevents a blunder and saves one reader some time, then it's been a success. But never fear; if you're like most users, installation will go smoothly and you'll enjoy the system more and more every day. Soon you'll cringe if someone asks you to use a PC without OS/2!

Bill Zinsmeyer is a programmer with many years of experience on mainframes who greatly enjoys using OS/2 while learning more about PCs. He can be reached on CompuServe at 71031,3436 and welcomes suggestions for future topics.

**Need more flexibility at the OS/2 prompt? More power in your batch files?
Then you need...**



JP Software's Enhanced Command Processor for OS/2

**4OS2 replaces the OS/2 command processor (CMD.EXE) with
a complete new interface, fully compatible with existing OS/2 commands.
It makes the [C:\] prompt friendly, powerful, and easy to use.**

- Full featured command line editing, command history, and recall.
- Over 40 new commands to make the command line a truly productive tool.
- Dozens of powerful new batch commands and functions, plus faster batch execution.
- Only \$89 bundled with 4DOS, JP Software's award-winning command processor for DOS (4OS2 only, \$69; \$29 for current 4DOS users).
- Aliases to create new or shorthand commands for common operations.
- Significant enhancements to most standard OS/2 commands.
- Complete, cross-referenced, on-line help.
- Shareware copies available on CompuServe (GO JPSoftware, library 10/JP Software) and many BBSes.
- Fully compatible with 4DOS and CMD.EXE, and with OS/2 versions 1.2, 1.3, and 2.0.

To order, call 800-368-8777 (order line), 617-646-3975, or 617-646-0904 (fax)

JPsoftware

P.O. Box 1470, East Arlington, MA 02174, USA

4OS2 is a trademark and 4DOS is a registered trademark of JP Software Inc. OS/2 is a registered trademark of IBM Corporation. Other company and product names are trademarks of their respective owners. Copyright ©1992, JP Software Inc.

Hot Off
The Press!

Develop GUI applications in COBOL
without learning "C".

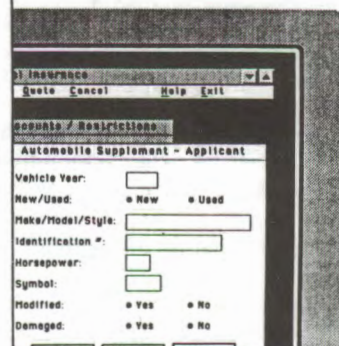
OS/2 Presentation Manager Programming for COBOL Programmers

Robert B. Chapman

OS/2 Presentation Manager
Programming for
COBOL Programmers



Robert B. Chapman



440 pages • Paperback with Diskette • ISBN 0-89435-439-6 • \$39.95

The first book in QED's OS/2 series teaches business application developers the techniques and basic skills necessary to create applications using IBM's OS/2 Presentation Manager. A major benefit of the book is to teach COBOL programmers to develop GUI applications without learning "C".

This book covers all aspects of PM from a simple, display-only window application to a multiple-window, DLL-supported, SQL-based, fully functioning business application. Topics include an introduction to current concepts and realities associated with programming in the OS/2 PM environment, an introduction to OS/2 PM, basic PM function calls necessary to initiate a PM application, and development and enhancement of a PM application.

Chapman includes a disk of object code, executable code, SQL, transaction input and output files, resource and icon files—everything covered in the book.

CONTENTS: Introduction. Presentation Manager. Create PM Environment and a Window. Display Information in a Window. Displaying Real Information. Menu Bar, Dialog Box, Icon, and Accelerator Table. Multi-line Entry, Listbox, Combo Box, Radio Buttons, Check Boxes. Modeless Dialog Boxes. Multiple Windows. Help Processing via IPF. Printing a PM Report. Introducing SQL I/O. Dynamic Link Libraries. Sample Programs Compile and Execute Environment. PM Subroutine Calls (The PM API). PM Messages, Styles, Classes, and Codes. Glossary. Index.



QED Publishing Group
P.O. Box 812070
170 Linden Street - Dept. C-137
Wellesley, MA 02181-0013

- ☐ Yes! Please send me _____ copies of *OS/2 Presentation Manager for COBOL Programmers* with diskette for \$39.95 each including shipping/handling. If I return the book(s) at the end of 15 days, I will get a full refund.
- ☐ Check here if enclosing check or money order. QED will pay the shipping/handling.
- ☐ Send FREE QED Book Catalog for other OS/2 books.

Name _____

Company _____

Address _____

City, State, Zip _____

Telephone No. _____

☐ Please charge my credit card. ☐ Visa ☐ MC ☐ AMEX

Card No. _____ Exp. Date _____

Signature _____

(Offer invalid without signature)

OS/2 + MIS

GARY MURPHY

Corporate MIS departments have been enamored with the idea of distributed computing for some time. Advances in hardware, networks and tools to manage distributed computing are reducing the risk of implementing client/server solutions in environments that are currently mainframe-centric. Once the decision is made to implement a client/server architecture in a corporation, data-processing professionals will be asked to help their management choose the right platform for their environment. Certainly, OS/2 Presentation Manager and DOS plus Windows will be two platforms which will be considered as possible candidates for deploying the graphical front-end.

Operational Stability

In the mid-1980's, both Microsoft and IBM recognized that the design of DOS was limited in its ability to support a heavily networked environment. Both companies also recognized the need for a better user-interface and the desire by individuals, namely in a corporate setting, to be able to have multiple applications available, perhaps all running at a given time. Applications were also growing. What once was seen as acres of memory, the 640K limit for DOS has quickly become to be seen as a barrier. OS/2 was designed to meet these emerging needs as the ideal multitasking client in a networked environment running large applications.

Deploying all or part of mission-critical applications on a desktop platform requires that the platform exhibit characteristics that are more similar to the mainframe architectures. The operating system must protect concurrently running applications from interfering with one another and keep an application from corrupting the operating system itself. To assure operational stability, the operating system must also be able to preempt a running application to service time-critical components. Finally, an ideal multitasking operating system would help ensure that the application with which the end-user is interacting provides crisp response, even under moderate to heavy load situations.

The design of OS/2 accommodates all of these needs quite well. Microsoft Windows version 3.0 has been a popular and successful operating environment for very good reasons. For true Windows applications, it has eliminated most of the limits associated with the 640K barrier of DOS. It provides a good, consistent graphical user interface to ease the learning curve when bringing in new Windows applications. However, Windows was built on the single-tasking DOS operating system and therefore is constrained by what it can do. Perhaps its greatest architectural weakness is that all Windows applications share the same address space. An errant application can potentially corrupt the data of another application, or corrupt the integrity of Windows itself, which results in the much-publicized Unrecoverable Application Error. Windows, version 3.1 has code that tries to keep applications from corrupting the Windows operating environment, but it

is still expected to offer little assistance in keeping one application from interfering with another.

Operational Synergy

The bulk of data processing done in most corporations is performed on IBM mainframes operating across SNA networks. When choosing the platforms on which to deploy early pilots for client/server computing, it is vital to consider the operational and support procedures within your organization. Most mainframe systems programmers have processes in place to collect problem source identification and pass it on to the vendor using such tools as IBMLink. The support for OS/2 and the Communications Manager and Database Manager components are handled much the same as their host counterparts in MVS, VTAM, and DB2.

A result of the design intent of OS/2 to support mission-critical applications in a networked environment is the availability of diagnostic tools. In the event of a component failure in the operating system, OS/2 provides dump and trace facilities that are not available with older generation desktop operating systems. These diagnostic tools contribute to quicker problem resolution time, which is essential in supporting a mission-critical application.

Investment Protection

If there is an embedded base of applications on the desktop in a corporation today, it is likely to be DOS or Windows, not OS/2 Presentation Manager. OS/2 version 1.3, provides only minimal support for DOS and Windows applications and is perhaps the greatest barrier to the acceptance of OS/2 by the end-users in many corporations.

The much-discussed OS/2 version 2.0 has the potential to overcome that barrier. Support for Windows standard mode applications and multiple applications is available in the OS/2 v2.0. In most environments, the Windows support in OS/2 will match or exceed the performance of Windows 3.0. The comparison, however, needs to be made to Windows 3.1. The performance of Windows 3.1 is expected to be better than that of version 3.0 and thus should provide a challenge for IBM to come close to its performance characteristics.

It should be noted that when looking at benchmarks, one should not rely on studies that compare a single task on native Windows with a single task on OS/2's Windows emulation. OS/2 has been shown to be able to better support a moderate to heavy workload environment typical of a corporate application engaged in client/server computing. If performance issues for Windows applications are of great concern, it is best for each company to benchmark in an environment that better simulates their anticipated workloads than to depend on published performance reports.

The other facet of imbedded base is the skill set of the end user. Again, if the end-user has experience with a graphical user interface, it will be Windows, not the OS/2 Presentation Manager.

Migration to the OS/2 Workplace Shell found in OS/2 v2.0 will require a new learning curve for its user base. The Workplace Shell is a next generation graphical user interface. It implements a more consistent and flexible way to manipulate the environment, which offers considerably more visual clues than in prior-generation user interfaces. When assessing the training effort, it is expected that the costs would be considerably less than the cost incurred in moving from the DOS command line to the Windows or Presentation Manager v1.x environments.

If the training costs become a barrier to implementing OS/2 v2.0, the Workplace Shell can be implemented to behave in a way similar to the Windows 3.0 or the Presentation Manager v1.x interfaces.

Native Program Development

If a company has chosen to develop the application using a third generation language and coding to the native API, OS/2 and the Presentation Manager offer a great deal more programmer facilities than the DOS and Windows APIs. The graphics subsystem in Presentation Manager, known as the Gpi, is significantly more sophisticated than the corresponding Gdi interface found in Windows. Interprocess communications facilities, such as pipes, queues and semaphores, are available in OS/2 to facilitate advanced applications design often found in client/server applications. There is no counterpart in the Windows environment unless the developer is willing to develop a substitute.

Another key area in the design of a client/server application is the support for multiple threads. The simplest client/server design would deploy a distinct thread for managing the host connection and another thread for supporting the presentation layer of the application. Since Windows does not support multiple threads, a developer implementing a client/server application would have to devise a work-around to manage both the presentation services and the connection to the remote data.

Development Tools

Due to the success of Windows in the marketplace, there are more fourth-generation developer's tools and user interface builders available for the Windows environment than the OS/2 Presentation Manager environment at present. This does not mean that the OS/2 PM environment is without a range of tools to implement client/server computing.

If there is a tool that meets the customers requirements and there is no corresponding tool in the Presentation Manager environment, then Windows needs to be considered for deployment platform for the front-end. Before making this decision, one must consider the incremental value of a Windows-based product over that which is available in the Presentation Manager environment and decide if that value exceeds the value of the increased operational stability and synergy of OS/2.

Windows NT and OS/2 Futures

Microsoft is developing a more robust operating system foundation, known as NT, for the Windows API that overcomes the limitations found in the present DOS environment. While the product is not yet available, it is currently being positioned by Microsoft as more of a server platform than the ideal client in a networked environment. Microsoft states that Windows NT will be available by early 1993. Until NT is more widely available to alpha and beta

testers, it is not possible to know if that projection is a reasonable one. For that reason alone, I would not recommend that a corporation make a business decision based on a product as immature as NT.

The recent partnerships formed by IBM with Apple, Motorola and others have cast a certain shadow of doubt on IBM's future intent for OS/2. The ultimate results of these business ventures is not known. However, IBM has publicly stated that whatever the result of these partnerships may be, there will be an upgrade path for OS/2 customers. Based on IBM's commitment to OS/2 in the past and present, it is likely that customers will have an upgrade path - at the source code level at the very least - to whatever future operating systems are developed in those spin-off companies.

Conclusion

The price-performance and the function of desktop computers create a compelling reason for companies to look at client/server computing. This requires that a decision be made on the desktop platform. The role of data processing professionals is to assist their managements in making an optimal decision. The needs of every corporation vary, but the approach to selecting the right choice is the same: Understand the arguments on both sides, weigh their respective importance to the organization and make the decision. In most companies with an investment in mainframes, OS/2 is the platform of choice. ♦

Learn OS/2 by Simply Watching TV!



As a leader in video training, we are proud to announce the addition of IBM OS/2 2.0 to our complete line of learner friendly videos.

Getting Started with OS/2, 2.0 \$39.95

This video tutorial will take the first-time OS/2 user through the many setup and installation options. Also learn the basics of the OS/2 environment.

Productivity with OS/2, 2.0 \$49.95

This professional training tool will increase productivity. Get the most from OS/2 using this clear, easy-to-follow video presentation.

(add \$5 for shipping)

**Get Both
for \$75**

**Call, send, or fax
your order to:**

31 S. Adair Street, Pryor, OK 74361
918/825-6700, 918/825-6744 (fax)
800/842-4723 (800/MAGRAP)

ViaGrafix
Computer Training Videos
"The Best in the Business!"

FOR IMMEDIATE RELEASE

World Wide OS/2 Developer Assistance —

IBM announced that it has enhanced the OS/2 Developer Assistance program (DAP) to reach software developers around the world who are interested in developing OS/2-based products. Developers can apply for DAP membership by calling 407-982-6408 to receive a sign-up form via fax, or by typing "GO OS2DAP" on CompuServe and completing the electronic sign-up form. In addition to current CompuServe support and country-specific electronic services — such as IBMLink in the U.S. — IBM intends to use CompuServe as one of the primary worldwide distribution sources for OS/2 product technical and marketing information.

IBM also announced that it will offer a beta version of OS/2 2.x code, beta version of various IBM OS/2 development tools and programming technical information on a CD-ROM that is being distributed in a pilot program. There will be a shipping and handling fee for the CD-ROM. Installation support will be provided to programmers via the CompuServe "OS2DEV" forum, section 16 "DEVELOPERS CD ROM." License for the code contained on the CD-ROM expires December 15, 1992.

The CD-ROM contains: OS/2 2.x 32-bit graphics engine beta; Windows 3.1 API support for win-OS/2; IBM developers toolkit for OS/2 2.0; Beta version of an enhanced IBM WorkFrame/2; IBM C Set/2 (latest Beta with VDD and new linker); profiler for C Set/2; MMPM/2 plus toolkit; NTS/2 beta code; Netware Workstation kit for OS/2 2.0; TCP/IP Ver 1.2.1 for OS/2; IBM OS/2 2.0 "Red Books"; IBM OS/2 2.0 Technical Library; 26 productivity tools (IBM employee-written software)

IBM, 1133 Westchester Ave., White Plains, NY 10604. Contact Rob Crawley, 914-642-5364.

Automated Test Facility (ATF 2.0)

Softbridge announces automated test facility (ATF 2.0). ATF is a software system designed for unattended testing of applications written under Windows, OS/2, and Dos, either standalone or in networked environments. ATF 2.0 provides powerful new features that increase the product's breadth while dramatically improving ease of use.

Softbridge, Inc., 125 Cambridge Pd Dr., Cambridge, MA 02140, 617-576-2257.

BenchTech for OS/2

BenchTech for OS/2 is a powerful performance measurement tool, containing a suite of 32-bit benchmarks designed specifically for OS/2 version 2.0 and beyond. OS/2 2.0 performance can vary greatly between one computer and another. Even though all systems that run OS/2 2.0 are based on INTEL 386 architecture, the observed performance is also dependent on the chase design

(if the system uses a cache), the number of wait states required, the video system, the disk drive and many other factors.

BenchTech for OS/2 allows you to measure integer and floating math performance, data movement and memory access. There are three separate disk tests, and you control whether the program uses the OS/2 system disk cache. The program allows you to determine relative video performance using any of eight video tests. Three application oriented tests are included or you can specify your own OS/2, DOS, or Windows test. Two higher level benchmark tests are included for comparing systems. Twenty-five benchmark tests are included in all.

Synetic Systems, 2210 Highway 6 & 50, Suite 204, Grand Junction, CO 81505-9406, 303-241-1717.

FactslinkPRO

FactslinkPRO development system is a comprehensive interactive voice/fax software package. It includes complete voice mail, fax mail, auto attendant, fax retrieval, database interaction and order entry capabilities in a single package. It runs with popular third-party computer fax board and voice board hardware. Applications can have up to 60 voice lines, and up to 30 fax lines per computer.

Voice Link, Inc., 1840 Oak Ave., Evanston, IL 60201, 708-866-0404.

CCC/Manager for OS/2 2.0

CCC/Manager for OS/2 2.0 is part of Softtools' CCC (Change and Configuration Control) product family, which provides extensive life cycle management functionality across all major platforms, including IBM, Digital, Unix and the PC. The product allows change and version control for all software components, support for all phases of life cycle, turnover and migration management, impact management, and cooperate across platforms through CCC/Bridge products.

Softool Corp., 340 S. Kellogg Ave., Goleta, CA 93117, 805-683-5777.

SizeIt

SizeIt, a new MS-DOS software utility that enables users to quickly and easily print ascii text files, using their printers built-in scalable fonts. Through SizeIt's menu driven interface, users select typeface, pointsize, page orientation, margins, headersize and other page layout and formatting options. This should be particularly interesting to Canon LPB-4 and LPB-8 printer users. However, many spreadsheet database, and other more specialized software applications still lack Canon support. Canon still has not yet delivered an OS/2 printer driver. This utility supports drag and drop printing now for plain text objects, allowing users to print ascii text any way they like, under OS/2. Succinct is now beta testing support for

the HP III series.

Succinct Systems, Inc., PO Box 281, Norwich, VT 05055, 802-649-5144.

OS/2 Compatible Tape Drives

Cristie, the data security company is launching two ranges of OS/2 compatible tape drives that can be connected by way of PC's Parallel printer port. This makes them easy to install and low cost to share. The TS4000 OS/2 and TS5000 OS/2 ranges include units with capacities from 60 M-bytes to 4.5 GBytes. It includes AT and MCA adaptor cards as well as the parallel printer port option. Each unit is provided with software to back up, restore and catalogue individual or groups of files. This can be operated via icons under Presentation Manager and is HPFS compatible.

Cristie Electronics Limited, Bonds Mill, Stonehouse, Glos. GL103RG, United Kingdom (045 382) 3611.

PM/Focus

PM/Focus offers OS/2 application developers a rich graphical toolset complete with file painter, forms painter, and Query Painter to build even the most complex applications without writing code. Application components such as databases, forms and procedures are transformed into simple graphic objects that can be moved, manipulated and activated with a point, click and drag of your

mouse. PM/Focus can take advantage of Information Builders' new Enterprise Data Access "EDA/SQL" technology. This means you can have access to more than 50 relational and non-relational file types on over 35 operating platforms.

A comprehensive set of list boxes, checkboxes, radio buttons, type fonts and other graphical images allows you to create attractive, intuitive applications user interfaces. PM/FOCUS OS/2 LANpak is available in 4, 16, 32 and larger user configurations.

Information Builders, Inc., 1250 Broadway, New York, NY 10001-3782, 212-736-4433.

SQA:Replay

SQA:Replay is a new utility that allows users to record every key-stroke, mouse movement and mouse click for playback at a later time. It is able to work in either full-screen character mode or with PM window, and no interference with user applications. Major replay features include the automatic restoration of the desktop order before replay begins, variable speed playback, the ability to selectively pause during playback, and unattended playback. It is upgradable to SQA:Robot, a full-featured test regression tool. SQA:Replay scripts can be used by SQA:Robot. SQA:Replay works under version 1.3 and 2.0 of OS/2.

Software Quality Quotomation, 1 Parker St., Lawrence, MA 01843. ♦

= PERFORMANCE 2.0

"Go as fast as 32 bits will take you"

A Tuning Kit for OS/2 2.0



PERFORMANCE 2.0 is both an instructional *booklet* and a *diskette* with a collection of program objects to help you tune your system.

THE BOOKLET - 60 pages - *the best source for OS/2 Tuning Information.*

BASIC section clearly describes a 32 bit, preemptive multitasking, multi-threading, demand paging virtual memory system, *written for the novice!*

DETAIL section focuses on those tip, techniques, and parameters that explain HPFS, FAT, Caching, Buffering, Swapping, Fragmentation, Multithreading, Timeslicing, and more - *written for the advanced user!*

THE DISKETTE - 25+ software OBJECTS - REXX source code included. These objects are designed to help you tune your system with SIMPLICITY.

OPTIMIZERS are the easiest objects to use. They assess your system, make recommendations and will perform a tune-up for you.

TOOLS are objects that tune individual parts of your system, such as Swap size, Buffers, Cache parameters, etc. Tools require some skills to operate.

ELIMINATORS are a reverse engineering of OS/2's selective install allowing you to selectively delete the files associated with a selective install.

All for \$19.95 + 2.50 S&H

send Check or M.O. \$22.45 to: **CLEAR & SIMPLE, INC**
or call (203) 658-1204 P.O. Box 130
VISA & MC Accepted W Simsbury, CT 06092

Ask about our new File Distribution System for Local Area Networks

OS/2 is a registered trademark of International Business Machines, Corporation

Peter D. Norberg Consulting

Professional Solutions to Professional Problems

- Convert DOS/Windows to OS/2
- DLL Design
- Devise Driver Implementation
- Expert C and Assembly Language Work

PDNVDISK -

The HUGE Ram Disk for OS/2

Call: (314) 647-8487

P.O. Box 9330 Richmond Hts, MO 63117

INTRODUCING CONTAINERS

ADVANCED PM PROGRAMMING

**GUY
SCHARF**

GUY SCHARF is president of Software Architects, Inc., 2163 Jardin Drive, Mountain View, CA 94040. Software Architects, Inc. specializes in OS/2 Presentation Manager software development and consulting. Guy can be reached on the CompuServe OS2DEV forum or on CompuServe Mail at 76702,557 or through Internet at 76702.557@compuserve.com.

Introducing Containers

OS/2 2.0 introduces four new controls: slider, value set, notebook, and containers. This month we will take a first look at containers.

First, what is a container? You have seen containers lots of them. The container control is one of the key components of the Workplace Shell. A container displays to the user the items, or objects, in contains in a visual way. Virtually everything you see on the OS/2 desktop before starting a program or opening a settings view is a container.

If you open the drives folder, select a drive, and open it, you will see a container showing directories on the drive, arranged hierarchially. This representation is called a tree view. Select Open from the system menu again, press the right arrow button, and you can open Settings, Icon view, Tree view, and Details view. These three views are all different ways a container can display the directories and files on a drive.

The container control supports five views of its contents:

Icon View Icons or bitmaps with strings beneath. The Workplace Shell desktop normally displays objects this way.

Name View Icons or bitmaps with text to the right.

Text View Text strings without any pictorial representation.

Tree View Icons or bitmaps with text to the right and with a way to display a hierarchy (often indicated by + and - signs).

Details View Icons or bitmaps, text strings, numbers, times, and/or dates.

Container windows support direct manipulation the user can select items with the mouse and drag them around. The user can drag items and drop them onto other containers, applications, or windows.

It is the container control that gives OS/2 2.0 and the Workplace Shell much of its flavor.

A container may change the way in which its contents are displayed by changing the view. In the Workplace Shell, different views are typically seen by opening windows with that alternate view. It is possible to change the view within a single window. Only one view can be displayed in one container window at a time.

A Step at a Time

As you might imagine, a control that performs all of these functions and many more and shows data in all of these views is complex. The documentation is daunting. The CUA Library/2 manual has 34 pages describing the container control and how it works and 74 pages of programming reference material. The documentation for the container control is larger than that for any of the other new controls.

Here, we will cut the problem down to a simpler case we will use the details view of a container to build a multi column listbox with scrolling, container and column titles, and other bells and whistles. Programmers frequently need multi-column listboxes for applications. While you can use the owner-draw feature of the listbox control to do that, the container with its details view is easier and more powerful.

Figure 1 shows a dialog with a container de-

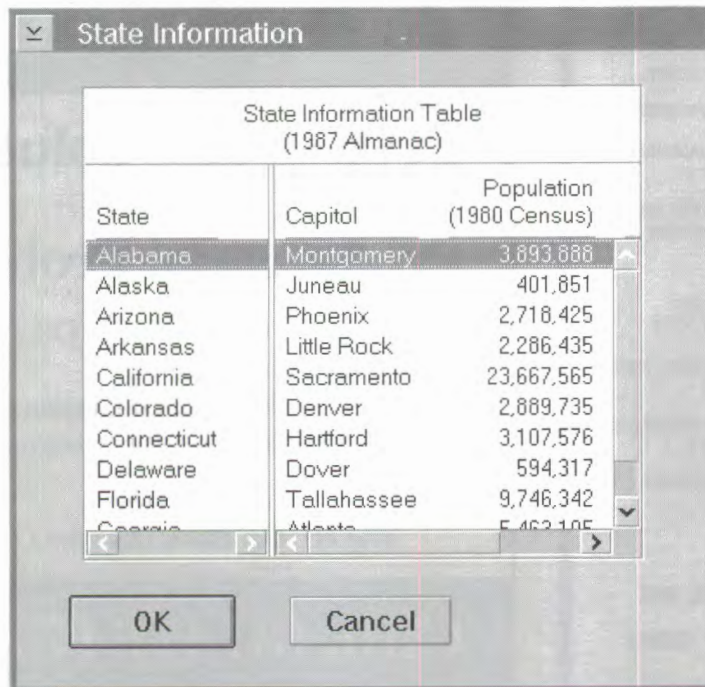


Figure 1

tails view that is displayed by the sample program. Look at some of its features that you could not easily achieve in a list box:

- The container has a two-line, centered heading that does not scroll. This heading is part of the container itself; it is not static text outside of the container. A horizontal line is placed beneath the heading.
- The columns have headings too. Some of the headings are left justified, some are right justified. These headings also have a horizontal separator line beneath them.
- The container has a vertical scroll bar. The scroll bar does not extend to the headings, indicating that the headings stay fixed as you scroll the container vertically.
- The container has a vertical double line between the first and second columns. Grab the line with the mouse and you can drag it left and right. This is called a split bar. The split bar separates the container into two pieces, each of which is individually scrolled horizontally. You can adjust the split of the window into two pieces by dragging the split bar.

If you scroll the container horizontally, the column headings scroll too.

- The data in the columns is formatted. Text strings are left-justified; numeric values and dates have correct punctuation as specified by the system settings and are right-justified.

This looks complex, but is all pretty easy to do. Let's first look at the basics of programming a container.

Container Basics

The basic steps for programming a container are the same as for other controls you are familiar with. You must create the container, load it with data, possibly monitor or modify its operation, possibly retrieve data from the container, and shut the container down when you are done with it. However, the container control requires more complex setup than older controls.

Creating a Container

As with other control, you create a container by placing a `CONTROL` statement in a resource script. `WC_CONTAINER` is the window class name (`CCL_CONTAINER` if you are using CUA Library/2). You can also create a container programmatically using `WinCreateWindow`.

In the sample program, the container is in a dialog box. In the Workplace Shell drive folders, the containers occupy the entire window. That difference is a design choice. For the sample program, I wanted the appearance of a list box in a dialog.

The container control supports a small set of style flags. We look here only at those that apply to the details view.

`CCS_READONLY` makes the entire container read-only. If you don't make the container read-only, the user can modify the container title and container records. If you want to modify some data, but not all, you would omit this style and later mark container titles, records, column titles, or column contents that are not to be changed as read-only.

`CCS_SINGLESEL` specifies that only one container item can be selected at a time. `CCS_MULTIPLESEL` allows the user to select zero or more items. Both of these are similar to listbox styles. The container control also has the `CCS_EXTENDSEL` style, which allows the user to select one or more container items.

`CCS_MINIRECORDCORE` specifies that a smaller record structure is to be used for this container. Records are a key concept of containers. We examine them in the next section.

Notice that the type of view was not specified as a style, and other information is not yet known either. You must initialize the container programmatically using the `CM_SETCNINFO` message. We'll look at that later.

Understanding Container Items & Records

A container item is anything the user or programmer might store in a container. The container is really a very general purpose object, and can hold many different types of items. Car, truck, bicycle, and unicycle objects could all be stored in the same container.

Programmatically, the data for each item in the container is stored in a record. The

container control manages the records. The programmer asks the control to allocate memory for the records, fills in the fields of the record, and tells the container to insert the records in the container. When the container window is closed, the container frees the memory for the records.

All records of a container must have the same structure. There are two issues in determining the record format:

- Whether the full-sized record format (`RECORDCORE`) or a small record format (`MINIRECORDCORE`) should be used.
- How much additional space should be included in each record for programmer-maintained data.

The basic record structure (`RECORDCORE` or `MINIRECORDCORE`) has control information used by the container to maintain the records.

The `MINIRECORDCORE` structure contains only the minimum information necessary for a container item: a pointer to a text string, a handle to an icon, a chain pointer to the next record, the position of the icon in an icon view, and some attribute flags.

The larger `RECORDCORE` structure contains the above information plus more. `RECORDCORE` supports different text strings for text, name, and tree views. `RECORDCORE` is the default data structure. If you wish to use `MINIRECORDCORE`, you must specify the `CCS_MINIRECORDCORE` style when creating the container.

The information in these data structures is sufficient to control the basic operation of a container, but is usually not sufficient for an application. Also, many of the items in the structure are pointers to strings. Where are the strings stored? If the strings are in memory the programmer has allocated, then that memory must remain allocated for the life of the record. That may not be convenient.

The container control allows you to append any information to the basic record structure. When you ask the container to allocate memory, you tell it how many extra bytes you want allocated in each record. The usual way to program this is to declare a C struct. The first element of the struct will be `RECORDCORE` or `MINI-`

INTRODUCING CONTAINERS

RECORDCORE. The following elements are your data fields. For example:

```
typedef struct
{
    MINIRECORDCORE    RecordCore;
        ULONG        ulMyNumber;
        CHAR        szMyString[10];
        CDATE        cdate;
} USERRECORD, *PUSERRECORD;
```

You ask the container to allocate memory for one or more records by sending a CM_ALLOCRECORD message. To be more efficient, allocate many records at the same time.

Now, complete the record contents. Assign values to elements in both the base record structure and the extra bytes that follow the base structure.

After completing the records, you insert them into the container by completing a RECORDINSERT structure and sending the address of it and the first record allocated to the container with a CM_INSERTRECORD message.

When the container terminates, it will free the memory it allocated for you. However, if you used malloc() to obtain any additional space while filling in the records, the container cannot release that space, since it does not know about it. Instead, you will need to examine the records, perhaps while processing the WM_DESTROY message to the dialog, and free any additional space you allocated.

You may remove or insert additional records in the container at any time.

Container Columns

The details view shows several columns of information. As you might expect, these columns require additional control information so that the container can manage the columns.

In a way similar to managing records, you ask the container to allocate column control structures with the CM_ALLOCDETAILFIELDINFO message. You then complete the FIELDINFO data structures with information about each column. Finally, you construct a FIELDINFOINSERT structure and insert the column information with a CM_INSERTDETAILFIELDINFO message.

As with records, you can remove or insert additional columns at will.

Unlike container records, the container does not allow you to add extra bytes to the column control data. Instead, the FIELDINFO structure contains a pUserData field that you can use to hold a pointer to your information.

Example Program

The example program constructs a details view container that lists the states of the Union, their capital cities, their population according to the 1980 census, and the dates they were admitted. The container can be scrolled vertically and horizontally. A split bar allows the user to adjust the amount of space in which the state name is displayed.

The Resource Script

Let's look first at the CNRDTL.RC file. As usual, all strings that the user might see are placed in a STRINGTABLE. No user-visible text is allowed in the program code. All of the strings are container or column headings. The \012 in the strings is a new-line character, forcing that heading to be displayed in two lines.

The dialog template contains a CONTROL statement for a window of WC_CONTAINER class. The user may select only one item at a time (CCS_SINGLESEL) and cannot modify any data (CCS_READONLY). The program uses the minimal record structure (CCS_MINIRECORDCORE).

A PRESPARAMS statement follows to set the font within the container to 8 point Helvetica. A \0 is included at the end of the font name string. This is required to make PRESPARAMS work with OS/2.0. This extra null is not required with OS/2 1.3.

The container control itself has no margin surrounding it. This makes it especially easy to use when the entire client area of a window is occupied by a container. We used the GROUPBOX statement to place a margin around the container for appearance's sake. This GROUPBOX statement should follow the CONTROL statement for the container itself.

Main Program

First, we define sample state data. Each record has the state and capital names, population, and date admitted.

Next we define the container record structure, RECORD. The first element of this structure must be MINIRECORDCORE (or

RECORDCORE if desired). We then add three fields that represent the columns to be displayed: a pointer to the capital name, population of the state, and the date admitted. We'll examine the data types and the following COLDESC data in the next module.

The main program simply runs the example dialog and terminates; it is of little interest itself.

Finally, we come to the dialog procedure, ExampleDlgProc(). The dialog procedure for this program is extremely simple. In WM_INITDLG, CnCreateDetailsView() defines the columns of the details view and InitContainer() loads the state data into the container. In WM_COMMAND, we dismiss the dialog when a button is pushed. In WM_DESTROY, CnDestroyDetailsView() frees memory allocated for the container.

Creating the Details View

CnCreateDetailsView() is a utility function in CNFUNC.C to initialize the details view of a container, based on control structures passed to it. This function requires the number of columns, a pointer to the column control structure, the column the split bar is to follow, the position of the split bar as a percentage of the container width, the STRINGTABLE id of the container heading, and a handle to the resource module containing the STRINGTABLE.

The column information is the COLDESC data in CNRDTL.C. The COLDESC data contains an offset to the data within the record, column attribute flags, a STRINGTABLE id for the column heading, an optional width for the column, and a NULL pointer used within CnCreateDetailsView().

The first thing CnCreateDetailsView does is go through the table of column descriptions, load the column titles from the resource file, and place pointers to the title text in the pszTitle field of the structure.

Next we initialize the container itself. We clear the CNRINFO structure on the stack and set the structure size in cb. We set the container attributes field in cnrinfo.flWindowAttr to CV_DETAIL (details view), CA_DETAILSVIEWTITLES (we want column titles), CA_CONTAINERTITLE (we want a title area for the container itself), and CA_TITLESEPARATOR (we want a horizontal line under the container title).

Then we load the container title from the

resource file and place a pointer to it in `cnrinfo.pszCnrTitle`. The title contains a new-line character (`\012`), which causes the title to appear as two lines.

The `CNRINFO` structure contains a great many fields. When we send the `CM_SETCNRINFO` structure to the container, we must tell it what parts of the `CNRINFO` structure should be examined. We tell it this with `CMA_*` flags OR'ed together in `mp2`. In this case, we say `CMA_FLWINDOWATTR` (set window attributes) and `CMA_CNRTITLE` (set the title text). Those are the only two fields in `CNRINFO` that we filled in. If we had completed other fields, we would have needed to specify other `CMA_*` values to ask the container to pay attention to those fields.

Now that we have established the container, we can define the columns. First, we use the `CM_ALLOCDETAILFIELDINFO` message to ask the container to allocate the column control structures for the number of columns we need. The container returns to us a pointer to the first `FIELDINFO` struc-

ture. The other structures have been allocated and chained to this structure. The structures have been initialized as zero except for the structure length and chain fields.

We now loop through the `COLDESC` array, filling in one `FIELDINFO` structure for each `COLDESC` entry.

We copy the field offset and the attributes from `COLDESC`. These attributes are key to the operation of the column. You must specify the type of field this is. Details view supports `CFA_ULONG` (offset is to a `ULONG`), `CFA_DATE` (offset is to a `CDATE` structure), `CFA_TIME` (offset is to a `CTIME` structure), `CFA_BITMAPORICON` (offset is to an `HPOINTER`), or `CFA_STRING` (offset is to a `PSZ`, not to a `CHAR` array as the name might suggest). We can specify the justification of the data (`CFA_LEFT`, `CFA_RIGHT`, `CFA_CENTER`), vertical positioning of data in the row (`DT_VCENTER`, `DT_TOP`, `DT_BOTTOM`), and visual attributes (`CFA_HORZSEPARATOR` for a separator below column headings, `CFA_SEPARATOR` for a vertical separator to right of column,

`CFA_INVISIBLE`, `CFA_FIREADONLY` for a read-only field, `CFA_OWNERDRAW`).

We don't use any bitmaps or time values in this sample. The `CDATE` and `CTIME` structures are sets of three `ULONG` values. The container will format them according to the current system preference settings.

We copy the title attribute and mark the title read-only. We use `CFA_LEFT` and `CFA_RIGHT` on columns to left or right justify the column headings. Since the container has the `CCS_READONLY` style, `CFA_FITTLEREADONLY` does not add anything. We have no user data and set the column width to 0. With a 0 width, the container will determine the width of the columns based on the data to be displayed. We could have set a width of our own choice if we wished to.

Setting the split bar requires modifying the container setup information. It is not part of the `FIELDINFO` information as such. We need to set a pointer to the last `FIELDINFO` that is left of the splitbar, and specify the position of the splitbar. To determine the position, we get the dimensions of the con-

Client/Server Programming with OS/2 2.0
2ND ED.

By Robert Orfali and Dan Harkey IBM Corporation
ISBN 0-442-01219-5
IBM Order No. G325-0650 \$39.95

"Professional programmers will be pleased with the meat in this book"
- Dr. Jim Gray in Database Programming & Design

THE DEFINITIVE GUIDE TO CLIENT/SERVER PROGRAMMING USING OS/2 2.0

This second edition contains over 600 new pages on advanced client/server technology and 32-bit OS/2 features. Through easy to read, in-depth tutorials and 300 pages of working code the book shows you how to integrate databases, LAN's and Workplace Shell clients to build client/server applications on an OS/2 platform.

The book covers OS/2 2.0's base services-including threads, semaphores and NamedPipes; LAN protocols - including CPI-C/APPC, NetBIOS, Sockets over TCP/IP, NetWare Requester, and LAN Server; and database servers - including Database Manager and DDCS/2. Over 200 pages are devoted to Object-Oriented User Interface (OOUI's) using SOM and the Workplace Shell Class Libraries.

The first 200 pages, "a book within a book", is a very readable, general introduction to the OS/2 2.0 Client/Server platform. The last 900 pages are great for programmers at every skill level.

Look for this book and other OS/2 2.0 Library Books from Van Nostrand Reinhold at Barnes & Noble, Bookstop, Bookstar, P. Dalton and other fine technical bookstores nationwide, or to order call our Toll-Free number 1-800-842-3636 Dept. Z1375



Publishing for Professionals Since 1848

INTRODUCING CONTAINERS

tainer window and calculate the percent of that width that was passed as a parameter. We set the calculated x position and the pointer to the last FIELDINFO in the CNRINFO structure. Finally, we send a CM_SETCNRINFO message to the container, specifying which fields we have set (CMA_XVERTSPLITBAR for the location, and CMA_PFIELDINFOLAST for the last column to the left of the split bar).

Having completed all FIELDINFO structures, we construct a FIELDINFOINSERT structure, tell it how many structures we want to insert, and insert the fields by sending CM_INSERTDETAILFIELDINFO.

Loading the Container

Now we are ready to load the container using InitContainer() in CNRDTL.C. First, we ask the container to create the records. The number of extra bytes is the difference between our RECORD size and the system's MINIRECORDCORESIZE. The container allocates the requested records, initializing

them to zero except for the size and chain fields.

All we need to do is to run the chain and our array of state data, copying information into the RECORD structure. We used the pszIcon field, which is the name of the item, as a pointer to the state name. We used the pszCapital name we created for the capital city. We could have used our own pointers for both. We could also have added CHAR fields for these names in the RECORD structure and copied the actual strings in. However, if we had done that, we would still have had to have PSZ values, as the CFA_STRING must specify the offset of a PSZ, not the beginning of a string.

Next we create a RECORDINSERT structure, give it the number of records, tell it where to insert these records, and insert the records with a CM_INSERTRECORD message.

We're done. The container is now up and running. Since this is a read-only dis-

play container, we don't have to do anything until the user terminates the dialog.

Shutting Down

While initializing the container and details view, we allocated memory with malloc. When the container closes, we need to free that memory to prevent a memory leak. Upon receipt of WM_DESTROY, the dialog calls CnDestroyDetailsView(). Here we free the strings loaded from the resource file. We read the CNRINFO data and free the container title text that we allocated. Next we free the column data and all records.

In this example, freeing the column data and records is not required, as the container will free that memory. However, if we had been doing something else, such as switching from a details view to a tree view, we might have wanted to free the column data before the container terminated.

Summary

There you have it a multi-column listbox with titles, column headings, vertical splits, and nice formatting. Later, we can add many more features direct manipulation (drag/drop), direct editing of column data, and more.

The programs in this article are available on the OS2DEV forum on CompuServe. GO OS2DEV and download file CNRDTL.ZIP from library 4, Ver 2.0.

**Is your LAN Serving you?
Or are you Serving it?**

**THE SOLUTION IS
Getting the right training from the start:**

- ▶ LAN Server 2.0 for Administrators
- ▶ Advanced LAN Server Performance Tuning

AR&D also offers *OS/2 programming classes in C, REXX, ObjectVision* and provides LAN Server installation support.

**Are you still
using SneakerNet?!!!**



**Classes on our site, or yours.
Call for registration information.**

Adaptive Research & Design Co.
12000 Biscayne Blvd., Suite 203
Miami, Florida 33181
Phone: (305) 899-0070
Fax: (305) 892-8669

MOVING?
PLEASE NOTIFY US
FOUR WEEKS IN
ADVANCE SO THAT YOU
DON'T MISS ANY
ISSUES OF
OS/2 MONTHLY . . .
AND DON'T FORGET TO
INCLUDE YOUR
ZIP CODE!

LISTING 1. CNRDTL.C

```

// cnrdtl.c - Sample program to demonstrate container detail
//
// cnrdtl.c
//
// Sample program to demonstrate container detail view
//
// By: Guy Scharf (415) 948-9186
// Software Architects, Inc. FAX: (415) 948-1620
// 2163 Jardin Drive
// Mountain View, CA 94040
// CompuServe: 76702,557
// Internet: 76702.557@compuserve.com
// (c) Copyright 1992 Software Architects, Inc.
//
// All Rights Reserved.
//
// Software Architects, Inc. develops software products for
// independent software vendors using OS/2 and Presentation
// Manager.
//
#define INCL_PM // Basic OS/2 PM
#define INCL_BASE // components
#include <OS2.H>

#include <stdlib.h> // C runtime library
#include <stddef.h>
#include <string.h>

#include "cnrdtl.h" // Container demo defs
#include "cnfunc.h" // Container utilities

// Sample data

typedef struct
{
    SHORT mm, dd, yy; // General date storage
} OURDATE, *POURDATE;

typedef struct
{
    PSZ pszState; // Name of state
    PSZ pszCapital; // Name of city
    ULONG ulPopulation; // '80 census population
    OURDATE ourdateAdm; // Date admitted
} STATERECD, *PSTATERECD;

STATERECD statedata[] =
{
    {"Alabama", "Montgomery", 3893888, {12, 14, 1819}},
    {"Alaska", "Juneau", 401851, {1, 3, 1959}},
    {"Arizona", "Phoenix", 2718425, {2, 4, 1912}},
    {"Arkansas", "Little Rock", 2286435, {6, 15, 1836}},
    {"California", "Sacramento", 23667565, {9, 9, 1850}},
    {"Colorado", "Denver", 2889735, {8, 1, 1876}},
    {"Connecticut", "Hartford", 3107576, {1, 9, 1788}},
    {"Delaware", "Dover", 594317, {12, 7, 1787}},
    {"Florida", "Tallahassee", 9746342, {3, 3, 1845}},
    {"Georgia", "Atlanta", 5463105, {1, 2, 1788}},
    {"Hawaii", "Honolulu", 964691, {8, 21, 1959}}
};

typedef struct // Container data record
{

```

```

MINIRECORDCORE RecordCore; // MINIRECORDCORE structure
PSZ pszCapital; // Capital city
ULONG ulPopulation; // Population
CDATE cdateAdmitted; // Date admitted
} RECORD, *PRECORD;

COLDESC cdState[] =
{
    {offsetof(RECORD, RecordCore.pszIcon), CFA_STRING | CFA_FIREADONLY |
    CFA_HORIZSEPARATOR | CFA_LEFT, IDS_HEAD_STATE,
    CFA_LEFT, 0, NULL},
    {offsetof(RECORD, pszCapital), CFA_STRING | CFA_FIREADONLY |
    CFA_HORIZSEPARATOR | CFA_LEFT, IDS_HEAD_CAP,
    CFA_LEFT, 0, NULL},
    {offsetof(RECORD, ulPopulation), CFA_ULONG | CFA_FIREADONLY |
    CFA_HORIZSEPARATOR | CFA_RIGHT, IDS_HEAD_POP,
    CFA_RIGHT, 0, NULL},
    {offsetof(RECORD, cdateAdmitted), CFA_DATE | CFA_FIREADONLY |
    CFA_HORIZSEPARATOR | CFA_RIGHT, IDS_HEAD_ADM,
    CFA_RIGHT, 0, NULL}
};

// Prototypes of procedures

static MRESULT EXPENTRY ExampleDlgProc (HWND, MSGID, MPARAM,
MPARAM);

static MRESULT InitContainer ( // Initialize slider
HWND hwndDlg, // I - Dialog window
USHORT idContainer, // I - Container id
USHORT cStates, // I - Number of states
PSTATERECD pastate); // I - State data array

//
//
// Main program to drive container example
//
//
int main (void)
{
    HAB hab; // Handle to anchor blk
    HMQ hmqMsgQueue; // Handle to msg queue
#ifdef OS220
    HMODULE hmodContainer; // Handle to cntr module
#endif

    hab = WinInitialize (0); // Initialize PM

    hmqMsgQueue = WinCreateMsgQueue (hab, 0); // Create msg queue

#ifdef OS220
    if (DosLoadModule (NULL, 0, CCL_CONTAINER_DLL,
    &hmodContainer))
        return FALSE;
#endif

    WinDlgBox (HWND_DESKTOP, HWND_DESKTOP, ExampleDlgProc, 0,
    IDLG_EXAMPLE, NULL);

#ifdef OS220
    DosFreeModule (hmodContainer);
#endif

    WinDestroyMsgQueue (hmqMsgQueue); // Shutdown
    WinTerminate (hab);
    return 0;
}

```


INTRODUCING CONTAINERS

```
//-----
//
// ExampleDlgProc() - Show state info
//
//-----

static MRESULT EXPENTRY ExampleDlgProc (
    HWND    hwndDlg,
    MSGID    msg,
    MPARAM    mp1,
    MPARAM    mp2)
{
    switch(msg)
    {
        //-----
        // Initialize dialog by defining the details view
        // in the container, loading records, etc.
        //-----
        case WM_INITDLG:

            CnCreateDetailsView (WinWindowFromID (hwndDlg,
                                                    IDCN_STATEINFO),
                                sizeof cdState / sizeof (COLDESC),
                                cdState,
                                0,
                                33,
                                IDS_TITLE,
                                0);

            //-----
            // Load the container
            //-----
            InitContainer (hwndDlg, IDCN_STATEINFO,
                          sizeof statedata / sizeof (STATERECD),
                          statedata);

            return 0;

        //-----
        // Process pushbuttons. They both just quit dialog.
        //-----
        case WM_COMMAND:
            switch (SHORT1FROMMP(mp1))
            {
                // Cancel pressed
                // Dismiss dialog
                case DID_CANCEL:
                    WinDismissDlg (hwndDlg, FALSE);
                    return 0;

                // OK button pressed
                // We're done
                case DID_OK:
                    WinDismissDlg (hwndDlg, TRUE);
                    return 0;
            }

            return 0;

        //-----
        // Recover memory allocated for the container
        //-----
        case WM_DESTROY:
            CnDestroyDetailsView (WinWindowFromID (hwndDlg,
                                                    IDCN_STATEINFO),
                                sizeof cdState / sizeof (COLDESC),
                                cdState);

            return 0;

        //-----
        // All other messages go to default window procedure
    }

    //-----
    default:
        return (WinDefDlgProc (hwndDlg, msg, mp1, mp2));
}

return FALSE;
}

//-----
// Function: InitContainer
// Outputs: none
//
// This function loads the container with all of the state data.
// See the article for a complete discussion of this function.
//-----

static MRESULT InitContainer (
    HWND    hwndDlg,
    USHORT    idContainer,
    USHORT    cStates,
    PSTATEREC    pastate)
{
    // Initialize slider
    // I - Dialog window
    // I - Container id
    // I - Number of states
    // I - State data array

    USHORT    i;
    ULONG    cbExtraBytes;

    // Loop counter
    // Extra bytes in record
    structure */

    PRECORD    precord;
    PRECORD    precordFirst;

    // -> container records
    // -> first container record

    RECORDINSERT    recordInsert;

    // Record insertion control

    //-----
    // Allocate memory for all user records
    //-----

    cbExtraBytes = (ULONG) (sizeof (RECORD) -
                             sizeof (MINIRECORDCORE));

    precord = (PRECORD) WinSendDlgItemMsg (hwndDlg, idContainer,
                                           CM_ALLOCORECORD,
                                           MPFROMLONG (cbExtraBytes),
                                           MPFROMSHORT (cStates));

    precordFirst = precord;

    //-----
    // Initialize all records
    //-----

    for (i = 0; i < cStates; i++, pastate++)
    {
        //-----
        // Initialize the container record control structure
        //-----
        precord->RecordCore.cb = sizeof (MINIRECORDCORE);

        //-----
        // Copy our data to the container record control struct
        //-----
        precord->RecordCore.pszIcon = pastate->pszState;
        precord->pszCapital = pastate->pszCapital;
        precord->ulPopulation = pastate->ulPopulation;
        precord->cdateAdmitted.month = pastate->ourdateAdm.mm;
        precord->cdateAdmitted.day = pastate->ourdateAdm.dd;
        precord->cdateAdmitted.year = pastate->ourdateAdm.yy;

        // Move to next record
        precord = (PRECORD) precord->RecordCore.precNextRecord;
    }
}
```



```

//-----
// Construct record insertion control structure
//-----

memset (&recordInsert, 0, sizeof (RECORDINSERT));
recordInsert.cb = sizeof (RECORDINSERT);
recordInsert.pRecordParent = NULL;
recordInsert.pRecordOrder = (PRECORDCORE) {(ULONG)
    MPFROMSHORT (CMA_END));
recordInsert.zOrder = (USHORT) CMA_TOP;
recordInsert.cRecordsInsert = cStates;
recordInsert.fInvalidateRecord = TRUE;

//-----
// Insert the records
//-----

WinSendDlgItemMsg (hwndDlg, idContainer,
    CM_INSERTRECORD,
    MPFROMP (precordFirst),
    MPFROMP (&recordInsert));

//-----
// Return to caller
//-----

return 0;
}

```

LISTING 2. CNFUNC.C

```

=====
// cnfunc.c - Container utility functions
//-----
//
// cnfunc.c
//
// Container utility functions
//-----

#define INCL_WIN
#include <os2.h>

#include <stdlib.h>
#include <string.h>

#include "cnfunc.h"

#define MAXMSGSIZE 255 // Max size of string

//-----
// UtilLoadString - load a string from the resource file
//-----

PSZ UtilLoadString ( // Load string from res
    USHORT usStringID, // I - String identifier
    HAB hab, // I - Current HAB
    HMODULE hmod) // I - Resource module or NULL
{
    PSZ pszBuf; // String buffer

    pszBuf = malloc (MAXMSGSIZE);

```

```

    if (WinLoadString (hab, hmod, usStringID, MAXMSGSIZE,
        pszBuf) == 0)
    {
        free (pszBuf);
        return (NULL);
    }

    return (pszBuf); // Return -> string
}

//-----
//
// CnCreateDetailsView
//
// Create the details view of the container. This requires
// setting up the column information.
//-----

USHORT CnCreateDetailsView ( // Create details view
    HWND hwndContainer, // I - container window
    USHORT cColumns, // I - Number of columns
    COLDESC acd[], // IO->column descriptor
    SHORT sLastLeftColumn, // I - Last column in
    // left split window
    // -1 means no split
    LONG lPctSplitBarPos, // Percent of cntnr width
    // to set split bar
    USHORT idTitle, // I - container hdg id
    HMODULE hmod) // I - hmod for resources
{
    USHORT i; // Loop counter

    CNRINFO cnrinfo; // Cntr info structure
    PFIELDINFO pFieldInfoHead; // -> First field
    PFIELDINFO pFieldInfo; // -> Current field
    FIELDINFOINSERT FieldInsertInfo;

    PCOLDESC pcd; // Column descriptor

    //-----
    // Get container text information
    //-----

    for (pcd = acd, i = 0; i < cColumns; i++, pcd++)
    {
        if (pcd->pszTitle == NULL)
            pcd->pszTitle = UtilLoadString (pcd->idTitle,
                WinQueryAnchorBlock (hwndContainer),
                hmod);
    }

    //-----
    // Initialize the container for a simple details view
    //-----

    memset (&cnrinfo, 0, sizeof (CNRINFO));
    cnrinfo.cb = sizeof (CNRINFO);

    // Set for Details View, with column titles, with icons and
    // not bitmaps, include a container title, with a separator
    // bar underneath it, and don't let the user change title
    cnrinfo.flWindowAttr = CV_DETAIL |
        CA_DETAILSVIEWTITLES |
        CA_CONTAINERTITLE |
        CA_TITLESEPARATOR;

```


INTRODUCING CONTAINERS

```

cnrinfo.pszCnrTitle = UtilLoadString (idTitle,
                                     WinQueryAnchorBlock (hwndContainer),
                                     hmod);

WinSendMsg (hwndContainer, CM_SETCNRINFO,
            MPFROMP (&cnrinfo),
            MPFROMLONG (CMA_FLWINDOWATTR | CMA_CNRTITLE));

//-----
// Get memory for the column information
//-----

pFieldInfoHead = (PFIELDINFO) PVOIDFROMMMR (
    WinSendMsg (hwndContainer,
                CM_ALLOCDETAILFIELDINFO,
                MPFROMSHORT (cColumns),
                0));

pFieldInfo = pFieldInfoHead;    // Start at top of list

//-----
// Load the field info structures
//-----

for (pcd = acd, i = 0; i < cColumns; i++, pcd++)
{
    pFieldInfo->cb          = sizeof (FIELDINFO);
    pFieldInfo->fldData     = pcd->fldAttributes;
    pFieldInfo->fldTitle    = pcd->fldTitle;
    pFieldInfo->pTitleData  = pcd->pszTitle;
    pFieldInfo->offStruct   = pcd->offField;
    pFieldInfo->pUserData   = NULL;
    pFieldInfo->cxWidth     = pcd->cxWidth;

    if (sLastLeftColumn >= 0)
    {
        if (sLastLeftColumn == i)
        {
            SWP swp;
            WinQueryWindowPos (hwndContainer, &swp);
            cnrinfo.pFieldInfoLast = pFieldInfo;
            cnrinfo.xVertSplitbar =
                (swp.cx * 1PctSplitBarPos) / 100;
            WinSendMsg (hwndContainer, CM_SETCNRINFO,
                        MPFROMP (&cnrinfo),
                        MPFROMLONG (CMA_PFIELDINFOLAST |
                                    CMA_XVERTSPLITBAR));
        }
        pFieldInfo = pFieldInfo->pNextFieldInfo;
    }
};

//-----
// Construct the FIELDINFOINSERT structure that describes
// the number of fields to be inserted, where they are to
// be inserted, etc.
//-----

memset (&FieldInsertInfo, 0, sizeof (FIELDINFOINSERT));
FieldInsertInfo.cb          = sizeof (FIELDINFOINSERT);
FieldInsertInfo.pFieldInfoOrder = (PFIELDINFO) CMA_END;
FieldInsertInfo.cFieldInfoInsert = cColumns;
FieldInsertInfo.fInvalidateFieldInfo = TRUE;

//-----
// Insert the fields.

```

```

//-----
WinSendMsg (hwndContainer, CM_INSERTDETAILFIELDINFO,
            MPFROMP (pFieldInfoHead),
            MPFROMP (&FieldInsertInfo));

return 0;
}

//-----
// CnDestroyDetailsView
// Destroy the details view of the container. This
// requires freeing column information.
//-----

USHORT CnDestroyDetailsView ( // Destroy details view
    HWND hwndContainer,      // I - Handle to cntr window
    USHORT cColumns,         // I - Number of columns
    COLDESC acd[])           // IO-> column descriptor
{
    USHORT i;                // Loop counter
    PCOLDESC pcd;            // Column descriptor entry
    CNRINFO cnrinfo;         // Container info structure

    //-----
    // Remove any column titles loaded into memory
    //-----

    for (pcd = acd, i = 0; i < cColumns; i++, pcd++)
    {
        if (pcd->pszTitle != NULL)
            && (pcd->idTitle != 0) )
        {
            free (pcd->pszTitle);
            pcd->pszTitle = NULL;
        }
    }

    //-----
    // Remove the column heading from memory
    //-----

    WinSendMsg (hwndContainer,
                CM_QUERYCNRINFO,
                MPFROMP (&cnrinfo),
                MPFROMSHORT (sizeof (CNRINFO)));

    free (cnrinfo.pszCnrTitle);

    //-----
    // Remove the column information of the container
    //-----

    WinSendMsg (hwndContainer,
                CM_REMOVEDTAILFIELDINFO,
                0,
                MPFROM2SHORT (0, CMA_FREE));

    //-----
    // Remove any records in the container
    //-----

```



```

WinSendMsg (hwndContainer,
            CM_REMOVEVRECORD,
            0,
            MPFROM2SHORT (0, CMA_FREE));

return 0;
}

```

LISTING 3. CNRDTL.H

```

=====
// cnrdtl.h - Definitions for cnrdtl demo

//-----
// Change the following as required for target system
//-----
#define OS220                // Define target system
//#define OS213              // OS/2 1.3 + CUA Lib/2

#ifdef OS220
    #define MSGID    ULONG        // OS/2 2.0
#else
    #define MSGID    USHORT       // OS/2 1.3
    #include <fc1cnrp.h>         // CUA Library/2
    #define WC_CONTAINER CCL_CONTAINER // Window class
#endif

// Defines for dialogs, controls
#define IDLG_EXAMPLE    100
#define IDCN_STATEINFO  200
#define IDS_HEAD_STATE  500
#define IDS_HEAD_CAP    501
#define IDS_HEAD_POP    502
#define IDS_HEAD_ADM    503
#define IDS_TITLE       504

```

LISTING 4. CNFUNC.H

```

=====
// cnfunc.h - Container utility functions

//-----
//
// cnfunc.h
//
// Container Functions
//
//-----
//
// Data structure used to describe field information
//-----

typedef struct                // Field (column) descriptors
{
    ULONG    offField;        // Offset of field in record
    ULONG    flAttributes;    // Field attributes
    USHORT   idTitle;         // Identifier of column title
    ULONG    flTitle;         // Title Attributes
    USHORT   cxWidth;         // Column width (0 = auto calc)
    PSZ      pszTitle;        // Pointer to column title text
} COLDESC, *PCOLDESC;

```

```

//-----
// Function prototypes
//-----

```

```

USHORT CnCreateDetailsView ( // Create details view
    HWND    hwndContainer,   // I - Handle to container window
    USHORT   cColumns,        // I - Number of columns
    COLDESC  acd[],           // IO-> column descriptor
    SHORT    sLastLeftColumn, // I - Last column in left split
    LONG     lPctSplitBarPos, // Percent of container width
    USHORT   idTitle,         // I - container heading id
    HMODULE   hmod);          // I - handle to resource file

USHORT CnDestroyDetailsView ( // Destroy details view
    HWND    hwndContainer,   // I - Handle to container window
    USHORT   cColumns,        // I - Number of columns
    COLDESC  acd[]);         // IO-> column descriptor

```

LISTING 5. CNRDTL.RC

```

=====
#include <os2.h>                // OS/2 definitions
#include "cnrdtl.h"            // Application defs

STRINGTABLE
BEGIN
    IDS_HEAD_STATE, "\012State"
    IDS_HEAD_CAP,   "\012Capital"
    IDS_HEAD_POP,   "Population\012(1980 Census)"
    IDS_HEAD_ADM,   "Entered\012Union"
    IDS_TITLE,      "State Information Table\012(1987 Almanac)"
END

DLGTEMPLATE IDLG_EXAMPLE LOADONCALL MOVEABLE DISCARDABLE
BEGIN
    DIALOG "State Information", IDLG_EXAMPLE, 40, 27, 200, 150,
        WS_VISIBLE, FCF_SYSMENU | FCF_TITLEBAR
    BEGIN
        CONTROL "", IDCN_STATEINFO,
            20, 31, 160, 110, WC_CONTAINER,
            CCS_SINGLESEL | CCS_READONLY |
            CCS_MINIRECORDCORE |
            WS_GROUP | WS_TABSTOP | WS_VISIBLE
            PRESPARAMS PP_FONTNAMESIZE,"8.Helv\0"

        GROUPBOX "", 300, 19, 30, 162, 116

        DEFPUSHBUTTON "OK", DID_OK, 15, 9, 48, 13, WS_GROUP
        PUSHBUTTON "Cancel", DID_CANCEL, 78, 9, 48, 13,
            NOT WS_TABSTOP
    END
END

```

LISTING 6. CNRDTL.MAK

```

=====
# Make File Creation run in directory:
# D:\P\MAGAZINE\CNRDTL;

.SUFFIXES:

.SUFFIXES: .c .rc

ALL: CNRDTL.EXE \
    CNRDTL.RES

```


INTRODUCING CONTAINERS

```

cnrdtl.exe: \
CNFUNC.OBJ \
CNRDTL.OBJ \
CNRDTL.RES \
CNRDTL.MAK
@REM @<<CNRDTL.@
/CO /M /NOL /PM:PM +
CNFUNC.OBJ +
CNRDTL.OBJ
cnrdtl.exe

;

<<
LINK386.EXE @CNRDTL.@
RC CNRDTL.RES cnrdtl.exe

{.}.rc.res:
RC -r .\$.RC

{.}.c.obj:
ICC.EXE /Ss /Kbgr /Ti /W2 /C .\$.c

!include CNRDTL.DEP

LISTING 7. CNRDTL.DEP
=====

```

```

# Make File Creation run in directory:
# D:\P\MAGAZINE\CNRDTL;

# Assumed INCLUDE environment variable path:
# C:\TOOLKT20\C\OS2H;
# C:\TOOLKT20\ASM\OS2INC;
# C:\IBM\INCLUDE;
# E:\GPF\INCLUDE;

CNRDTL.RES: CNRDTL.RC \
# {$(INCLUDE)\OS2.H \
# {$(INCLUDE)\CNRDTL.H \
# {$(INCLUDE)\FCLCNRP.H \
CNRDTL.MAK

CNFUNC.OBJ: CNFUNC.C \
# {$(INCLUDE)\os2.h \
# {$(INCLUDE)\stdlib.h \
# {$(INCLUDE)\string.h \
# {$(INCLUDE)\cnfunc.h \
CNRDTL.MAK

CNRDTL.OBJ: CNRDTL.C \
# {$(INCLUDE)\os2.h \
# {$(INCLUDE)\stdlib.h \
# {$(INCLUDE)\stddef.h \

```

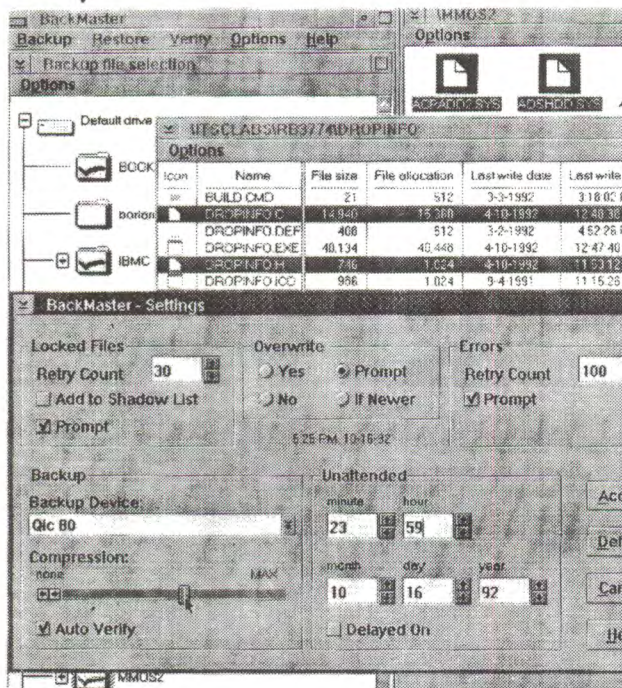


BackMaster

CHRISTMAS
SPECIAL

\$49.95
Until
Dec 25

OS/2 2.0 32-Bit PM Based Backup Utility.



Supports FAT, HPFS, Long Names, and Extended Attributes.
Variable Data Compression, Printed and On-Line Documentation.
Backup/Restore to Floppies and QIC-40/80 floppy Tape Drives!
Incremental Backups: Wildcards, Time/Date, & Graphical Selection.
We Support CMS JUMBO Tape Drives.
Unattended / Delayed Operation...
Runs Transparently in the Background.
Exploits Multithreading for better performance.
List Price \$79.95, Special \$49.95 + \$5.00 Shipping until Dec 25.
Texas Residents add 7% Sales Tax (\$3.49).
For More Information Call or leave a message on our support BBS.
Call our 24 Hour Support BBS 2400-14,400bps, for Demo Version.
To Order; Call, or Send Check, money order, VISA/MC to:

MSRDevelopment

Rt 7 #6409 Nacogdoches Tx, 75961
Ph (409) 564-1862 --- BBS (409) 560-5970

PAINT BY NUMBERS

THE ULTIMATE OS/2 GAME

**TIMUR
TABI**

Timur Tabi can be reached at 2 Lloyd Haven Drive; Huntington, NY 11743. Source code is available on the OS/2 Shareware BBS, 703-385-4325.

I love writing this column. Not only do I look forward to actually playing this game once it's done, but I like contributing to something in which I strongly believe, and I really do believe that OS/2 is the wave of the future.

Time constraints have prevented me from adding as much as I promised last month. The only new enhancements you will find are the implementation of dialog boxes for the "Map" menu items. Do not underestimate the amount of work required to support these features, however. A whole module was added for the load and save routines. The basic skeleton of the code comes from Petzold's book, an antiquated yet still valuable resource.

The dialog boxes themselves are created with the Dialog Box Editor that comes with the tool kit. The editor creates a .dlg, a .res, and a .h file. The .dlg file contains the resource script commands that define the dialog boxes. The .res file is a compiled version of the .dlg file.

There are a few things that should be kept in mind. First, do not include .res file as a module

in the make file because the resource compiler (rc) cannot combine individual resources. Every time rc links a resource file into the executable, it replaces the existing resources with those in the .res file. If you want to include several types of resources, then you must create a master resource script file which contains "rcinclude" statements. This process will create a single .res file containing all your resources.

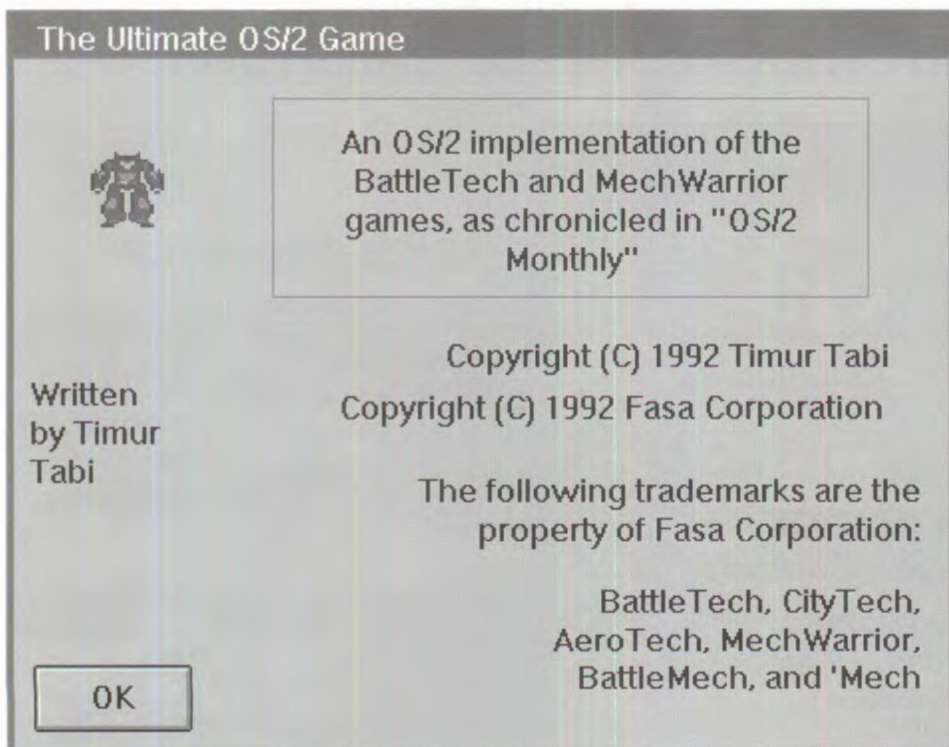
Secondly, beware of errors in the online help for the tool kit. The only way to be certain of the information is to check the header files. An example is the structure definition for FILEFIND-BUF3. Compare this with the entry in bsedos.h and you will see what I mean.

Last of all, double-check all the parameters when using code designed for OS/2 1.x. Most of the 16-bit integer parameters are now 32 bits long. Note that there is no speed difference between short and long integers on a true 386/486 system, since the 32-bit wide data bus can swallow both sizes in one gulp.

Look at the file game.rc, which is the master resource file for this project. It includes the dialog.dlg file that the editor creates. Note that both files include dialog.h - the header file also created by the dialog editor. I am not sure why this is necessary, but hopefully an explanation will present itself in due time.

The dialog box for loading and saving files follows a convention that exists in most applications. The two list-boxes show a list of directories and a list of files in the current directory and an entry field allows you to type in a file or path name. Selecting a file from the list-box places that name in the entry field automatically. Once you have selected a file, click on "Save" to save the current map to that file, or click on "Load" to load that file. The current implementation is not 100% complete. The load and save option should not be combined, and there is no check for overwriting an existing file. These changes will be available in the next installment.

The code for controlling both list-boxes is in the file files.c, which will eventually become a module of generic file I/O routines. The fill-functions for the list-boxes are very similar. Any existing entries are erased and



TIMUR TABI

the new entries are determined by scanning the current drive and directory. You may notice that the list-boxes are extra wide. This allows most HPFS long filenames to be displayed in their entirety. I only wish that my other applications worked the same way. As with everything else in this project, the dimensions of the dialog box can be changed with the editor to suit your preferences. Those of you with high resolution monitors and tiny system fonts will probably want to increase the size even further.

Notice that the DosFindFirst/Next functions ignore any extended attributes. The map files will eventually be given a file type (a.k.a. association) that will distinguish them from other files without needing a special extension. The short-term goal of this project is to bring the game up and running as quickly as possible, so that no one has to wait until 1994 before he can play. Note that I did not say 1993. Sorry folks, but this thing will not be done before Christmas — it should be ready by spring.

My roommate Scott and I got into a dis-

cussion over the importance and design of a user interface. We ended up agreeing on almost everything, but the differences between the programmer (me) and the user (Scott) became very apparent. Scott argued that the UI should be designed and coded at the same time as the rest of the program. His concern was that leaving the interface until later would cause inconsistencies between the interface and the application. I had to agree with him on this point, but I also had to mention that trying to create both at the same time causes most programmers to bite off more than they can chew.

Frankly, I think that attempting to fully design an application, interface and all, before any coding is started is a pipe dream that only works with software devoid of creativity. Remember, this game cannot be designed beforehand because it is a group project. New ideas and changes occur constantly, so the primary goal for the present is not to design a complete user interface, but rather a flexible one. What does that mean, you ask? It means that the applica-

tion should not be designed with only one "look-and-feel" in mind.

In other words, there is more than one way to skin a cat, just as there is more than one way to create a playing field. Pull-down menus allowed me to create a map editor which worked without wasting too much time on the code. Had I taken the time to create something more sophisticated, it would have been vastly more time-consuming to modify the more complicated code. And that is only the short-term problem. In the long run, we would probably end up with a user interface which no longer suited the task at hand. Therefore, our approach will be to formulate an interface which expands along with each new feature.

This, in a nutshell, is my philosophy on user interface design. I will be very happy to entertain any comments on the subject. Scott, by the way, is responsible for the new icon, which is much better than the one I designed. If anyone thinks he or she can do a better job, feel free to submit your creativity. ♦

OS/2 "BATCH MANAGER"

TO SUBMIT, MONITOR AND CONTROL BATCH JOBS

Batch Manager					
Queue Job Help					
Job ID	Job Name	Start Date	Start Time	Status	Single/Recurring
BATCHQ1		Active			
11	SALES ORDERS	04-22-1992	08:32	Active	S
7	Daily DTL upgrade	04-23-1992	07:00	Delayed	R
9	DAILY BACKUP	04-22-1992	23:30	Delayed	R
8	PAYROLL	04-30-1992	12:00	Delayed	R

CUA COMPLIANT

\$100 PER OS/2 WORKSTATION

For Information or To Order, Contact:

M. BRYCE & ASSOCIATES, INC.

777 Alderman Road

Palm Harbor, FL 34683

Tel: 813/786-4567

Fax: 813/786-4765

JOB SUBMISSION

Batch Manager Job Submission/Update

ENTER JOB NAME:

ENTER COMMAND FILE NAME:

OPTIONAL PARAMETERS:

Current Directory: C:\

FILE	DIRECTORY
STARTUP.CMD	A:] B:] C:] D:]

☐ SINGLE JOB SUBMISSION

START DATE:

START TIME:

☒ RECURRING JOB SUBMISSION

OK CANCEL HELP

START TIME:

BUILDING PROGRAMS

OBJECT OBJECTIVE

**DAVID
MOSKOWITZ**

If you've read the past articles in this series, you know we've devoted lots of space to a discussion about what an object is and why it makes sense to consider changing your development model to use an object-oriented approach. We've discussed the need to develop a different mindset for object-oriented development compared to classic procedural development. In addition, we've talked about OOP being an iterative process. There are a couple of conclusions that might not be obvious if you take these ideas to their extremes.

When we build object-oriented programs, very often we aren't really building programs, per se. Instead, we define objects and build the functionality of the completed program by arranging for objects to collaborate with each other. In an object-oriented development environment, rather than think about the final completed program, we concentrate first on building objects that can be re-used. Applications are really objects combined to produce results.

In procedural code, we build applications out of cooperating functions. We use that knowledge when we design and build systems. For example, we know the calling conventions of the public functions. Sometimes, we might even know something about the data structures used by the functions. (It is interesting that ever since David Parnas's article "On The Criteria to be Used in Decomposing Systems into Modules," appeared in the Communications of the ACM in December 1972, we've had formal "proof" about the problems associated with an approach to development that ignores hiding implementation details. If you look at code, today, you can still see global data, etc).

Object oriented design

We build object-oriented applications out of collaborating objects. Each object has its own set of responsibilities. The object's responsibilities are encapsulated with the class definition. Each object has a defined set of behaviors. Further, the only thing one object "knows" about another object is the operations that the other object can be requested to perform.

To put it another way, an object only knows about the messages it can send to other objects.

Messages are requests that an object respond. The requesting object does not know, nor does it care how the request is handled.

The published list of operations defines the responsibilities of a class of objects. As we design object-oriented applications, our first step is to identify the objects. As we identify objects, we also have to define the responsibilities or behaviors.

During the early stages of object-oriented design, we are less concerned about "how" any particular class of object will behave. Instead we concentrate on "what" the object does.

This level of abstraction is important. During the initial stages, we are not concerned with the internal logic that defines "how" the object behaves. Instead we concentrate on a definition of what the object can do, what responsibilities the object has and what services it may require from other objects.

Once we've identified an object, we decide what behavior other objects will require of this object (its responsibilities). The actual logic or internal behavior encapsulated (or hidden) within the object, itself. The process of object-oriented design is, in part, based upon the difference between the public information about the behavior of the object and the internal private logic that makes it work.

Basically, object-oriented design means we have three tasks. First, we have to identify the objects. Fundamentally, this can be viewed as an English exercise. Find the nouns or noun phrases in the specification and assume that as a starting point for object recognition. (We group the objects by similarities and differences that this helps us identify classes.)

Once we have the list of object candidates, we turn our attention to object behavior. We use the *behavior* of the objects to build the conceptual model of the application. It is the object's *behavior* that we map to become the responsibilities of the object. We ask ourselves, "What steps, actions, or results does this object produce?" Notice, this is a "what" question, not a "how" question.

After we've identified the objects, and responsibilities, we begin to ask which objects handle which tasks. Which other objects are necessary to complete the given, "Assignment?"

A line from an old Jefferson Starship song was

David Moskowitz is president of Productivity Solutions, 164 Avondale Road, Norristown PA 19403-2952. The consulting firm specializes in OS/2 and object oriented development. He is the author of *Converting Applications to OS/2*, Brady Books, * 1989, and the forthcoming revision. He can be reached on CompuServe at 76701,100, on MCIMail at 341-4084.

DAVID MOSKOWITZ

true about people and true about objects, "No man is an island, he's a peninsula." No one object will provide all the responsibilities necessary to completely map a complex application. Objects collaborate with each other; or, if you will, they delegate responsibilities to other objects.

So, the next phase of object-oriented design is to identify what this object needs from other objects, and which other objects (or collaborators) must be part of the system.

In effect, we define the public interface that "documents" the responsibilities this object accepts. Based upon this knowledge, we can select other objects to collaborate with this object to accomplish a specific function.

CRC

There is a name given to this approach of object-oriented design — CRC cards. CRC cards are standard lined index cards (either 3 by 5 or 4 by 6 depending upon how small you write, or how much you want to put on a card). CRC stands for Class, Responsibilities, and Collaborations.

Draw a line about one-third the horizontal distance from the right (on a 4 by 6 card, draw a line 4 inches from the left side). Put the name of the class on the top line of the left hand side of the card. Use the left hand two thirds for the responsibilities of the class (and object members of the class). In the right third, place the collaborating classes (or objects).

That's it — one class per card. It doesn't make any difference how many cards you have. It is better to have too many classes than too few; it is easier to prune things you don't need than it is to add things you've forgotten.

If something comes up either in discussions with co-workers on during your own work with the problem, you can always add a class. (CRC cards can be passed around a small implementation group for comment, review, etc.) Similarly, you can discard cards that no longer seem appropriate in whatever new context evolves as you work.

Don't just stand there....

Once you have a start on design, begin to implement the classes. Don't wait until the design is completed before you start implementation (i.e., building objects). Object-oriented development encourages the design a little code a little approach. The only way you'll get comfortable with the methodology is to use it.

Object oriented development also allows us to make mistakes. We always learn more about a system as we play with it. We always see things we could have improved or will improve next time. Object-oriented development is the way around the old classic way to create systems — "Build the whole, thing, push it off a cliff, watch it crash, and start all over." (I've heard this referred to as the Wright Brothers approach. However, Orville and Wilbur didn't do that.)

P-s-s-t !

No need to tell anyone you have CommandLine. Tell your friends you aren't fazed by all the icon clutter - 'Icons! Love 'em! Nothing like a little Shredder Action ... Mouse freak? Hey, I wish they made a FOUR button mouse! Don't know why they call it a Workplace; it's a fun place to me". And with CommandLine, it certainly will be!

**Command
Line™** 32-bit
Version 2.0
for OS/2 2.0

- Intelligent Application Launching with:
- Hotkey Application Start, Bait & Switch
 - Command Aliasing & History
 - Inline File Find & Filename Completion
 - Alternate Command Shell Support
 - Rexx Expression Interpretation
 - Instant WPS Object Creation
 - Active Window Toggle
 - Ideal for portables

\$39⁹⁵*

Because APIs DO fail . . .

We all know that Toolkit APIs fail. So that you don't have to check every call, Error Manager does. EM tells you which API failed, where it failed and why. EM can log all errors to a file or pipe, and optionally notifies your window procedure enabling centralized error processing.

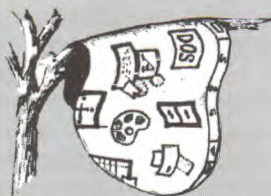
**Error
Manager** 32-bit
Version 2.0
for OS/2 2.0

- Invaluable for Debug, Test and Production cycles.
- Finds intermittent errors without the Debug Kernel.
- Includes pointer validity subroutine.
- Requires no assembly knowledge.
- Works with Optimized code.

List \$225
Introductory Price: *

Include 1 header file.
Link 1 DLL.
Set 1 environment variable.

\$175



Soft & GUI

2224 East 21st Street
Brooklyn, New York 11229

(718) 769-8017

* Plus \$5 Shipping & Handling. NY State residents must add sales tax.

The development loop

The key point about object-oriented development is that it does not progress in neat, nicely-bound stages. It is an iterative process. If you discover you need a new object, add it. If you picked an incorrect object, discard it. If you didn't get it exactly right, change it.

Object-oriented development supports change in a way that is very different from classic structured programming. Object-oriented development encourages change as we learn more about the system and its needs. Using structured programming, we had to specifically design change points into the code. We had to document all the interactions (well, that was the idea, on a large project (i.e., more than three to five people) trying to document the dependencies either never happened, was incomplete, or wasn't kept up to date).

Using an object-oriented approach to design empowers us to make mistakes, to make changes, to start with something that works and grows, to learn more about the system as we build it. This is the real power of object-oriented development. As we design object-oriented systems, we use the inherent isolation of each object to our advantage. Since the only thing published about an object is what it does, we are free to change the internal representations, fix bugs and extend the objects' capabilities without any risk of introducing bugs in other parts of the system because of interaction. In fact, it is this lack of side effects of changes that is another way of describing object re-use.

Client-Server revisited

One last point, when we talk about collaboration in the object-oriented world, we aren't talking about anything new or totally unique to OOP. For over three years, we've been telling developers to use the client-server model to build single user OS/2 applications.

The usual context for the discussions revolved around the use of OS/2 dynamic link libraries (DLLs). The object, was to recognize that at any point in time, any given DLL could be a server to one section of code and a client of another. A DLL isn't always intended to be a "server" or a "client." Instead, client and server are roles that dynamically change while the program runs.

In an object-oriented world, we still use the client server concept. A collaborating object may be a server to the requesting object. At the same time, it can be a client to a third object. We don't have to assign a DLL to be either a client or a server. However, the relationship between client and server is important.

If it proves helpful, you can visualize client-server as if a network is involved. The client makes a request of the server, over the network. The client cannot get access to anything the server doesn't explicitly send to him since the two are running in two separate computers. You can think of object collaboration in this way, whether or not a network is involved.

Coming attractions

Next month we'll take look at design a bit more. In addition, we'll also begin to talk about picking object-oriented tools and languages.

Windows, OS/2 & Unix

Consultants Programmers Systems Analysts



800-862-2211
Ext. 50 #

Technologies, Inc.

We provide on/off site consultants and programmers for short or long term assignments, who are specialists in, but not limited to, application design and development in the following areas:

Microsoft Windows
Presentation Manager
Communication Manager
Database Manager
LAN Manager
X-Windows
UNIX Communications

Some of the services we can provide you with:

Product Planning
Cost Analysis and Projections
Project Leadership and Management
Requirement Analysis
Functional Specifications
SAA/CUA Application Design
System Testing and Verification
System Design and Development
Host Front-end Design

Come to us first when you need software engineers to design, code and implement your Microsoft Windows, OS/2 and Unix applications.

Ceres Technologies, Inc.

8903 Glades Road Suite L-9234 Boca Raton, FL 33434-4021

IN THE TRENCHES

To NT, or not NT, that is the question... Whether 'tis nobler to suffer the slings and arrows of outrageous propaganda.... When I hear people talk about Microsoft's NT operating system, my first response is "What new technology?". NT is based largely on the Mach kernel, and having OS/2 source code taught the boys from Redmond a lot. Oh, sure, there's some neat new file system stuff, but to hear it tell from the Windows bigots, NT is a revolution that is so much better than anything else out there. I guess these guys have never heard of Unix.

But the really amazing part of all this is how these guys compare NT's proposed features with what OS/2 has now. I have news for all those who wait with baited breath for NT. It's going to be late. Like 2Q93 late (although the 'Softies are probably aiming for Spring COMDEX in May).

How concerned is MS about bugs and problems? Well, they're even thinking of calling the OS Windows NT 3.1. Now I don't know about you, but anyone who would be taken in by such a stupid ploy deserves what they get. Like the fact that there are whole classes of software that will not work under NT (those that access the hardware directly).

NT was originally designed as a server OS, which is why the security issue is so important. People expecting to run all their DOS apps without changes are going to be in for a rude shock. Further, NT's micro-kernel approach is less speedy than a traditional OS like OS/2. Of course, MS wants you to run NT on fast CPUs anyway, like RISC machines. OS/2's superior DOS compatibility is a strong reason why it is so popular. NT's DOS compatibility will be weaker, thus hurting its chances in the retail channel.

Windows bigots do not understand that the competition for OS/2 2.0 is not NT, it is DOS/Windows. MS does not plan to put NT on every desktop (not every machine has the oomph for NT). And IBM does plan to make their own microkernel based version of OS/2 (called Portable OS/2). And unlike MS, IBM is using its existing software and expertise (in conjunction with Carnegie Mellon University) to provide a new product. MS wants to re-invent the wheel, it

seems. Most likely Portable OS/2 will be merged with IBM's AIX (Unix-based) product. Look for an announcement on this at Fall COMDEX.

Consider the Win32 and Win32s APIs. Why are so many rushing to develop for NT, which hasn't proved it can sell worth a damn? To take advantage of NT, you have to make a program that runs only on it. If you make a 32 bit Win32s program to be compatible with Win 3.1, it will be creamed on NT by its feature-rich competitors. I think these developers are being suckered into making software for an unproven market so MS can hype up NT applications availability.

Me, I believe you should develop for existing markets first, then developing markets



second. Right now, with over a million copies of OS/2 being sold, this is where the action currently is. And if porting to NT is as easy as they say, it makes sense to wait until NT takes off (assuming it does).

But there is an even more crucial problem facing Microsoft and NT: the issue of support. If NT is to be a server OS, and is to make it in the Fortune 500 world, it will have to provide support. This means 24 hour a day, worldwide support for its product. And frankly, MS doesn't have the oomph to do it. One thing about IBM's bloated bureaucracy, there is an office almost everywhere you go.

I think NT will find some acceptance, but it will never achieve the market penetration that OS/2 does. New users of 486 machines might go for it. And of course, the die-hard Windows bigots. But to listen to the Windows crowd, they seem to think that NT will make OS/2 dry up and go away. Sorry, Bill.

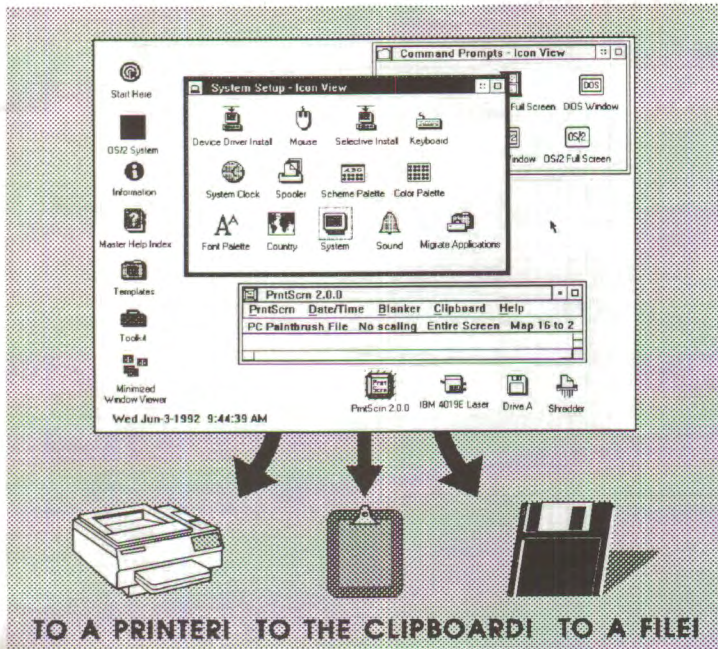
By the time NT ships, OS/2 will have had a lot of bug-fixing and cleaning up, as well as the necessary drivers (about time!) for video cards and other peripherals. Further, there will be a host of 32 bit OS/2 apps from companies like Lotus, Borland, and WordPerfect. When NT does hit the streets, their users will have to go through the same painful process of waiting for bug fixes, applications and so on that we OS/2 users have had to put up with. By the time the end of 1993 rolls around, NT will have finally started to pick up steam. But OS/2 will have had over a year long head start.

Finally, it's easy to show the advantages of 32 bit OS/2 vs 16 bit Windows to users. Try selling an end user on the idea of having C2 security. This is something for LAN administrators.

Now, I'm not saying NT will be a disaster. It will have its place, just as OS/2 and Windows 3.1 will. But for now and the foreseeable future, the OS of choice will continue to be DOS for the majority of users. New users with powerful machines will probably not choose a DOS-only platform, but there are still a lot of 286 and XT boxes out there. Eventually, though, I believe OS/2 will grab a larger market share than Windows (whatever flavor). Of course, this assumes that IBM actually wakes up and listens to its customers about how to market and improve its product. If not, well, the game

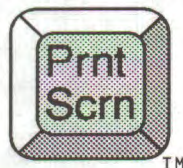
Eric Pinnel

CAPTURE YOUR SCREEN



MITNOR Software
28411 E. 55th Street
Broken Arrow, OK 74014

Phone: (918) 357-1628
Fax: (918) 357-2869



PrntScrnm™ is THE One-Step Screen Image Utility for copying a graphics image from the Workplace Shell screen to a LAN or locally attached printer, to the Clipboard, or to a disk file. The screen images appearing in this publication were captured and processed using the **PrntScrnm** utility. The capture operation can be initiated from the keyboard (by pressing the print-screen key), from the mouse, or from another program. The captured image area can be the entire screen, current active window, current active client area, selected window, selected client area, or specified rectangular area. Disk file formats include PM Metafile, PM Bit Map, PCX, TIFF, GIF, and WPG. The "Color Mapping" feature provides the control needed to produce quality printed images from color screen images. **PrntScrnm** has clipboard Viewing, Printing, Exporting, & Importing capabilities for Bit Maps, Metafiles, and Text files. **PrntScrnm** includes a Screen Blanker and a digital Date & Time "information bar". **PrntScrnm** is priced at \$115 and is available from major software resellers, or contact MITNOR Software.

RimSTAR PM Editor

**Special
\$99
Offer**

Do you need an Editor that has ... ?

Presentation Manager support: Full SAA/CUA PM Multi Document Interface. Complete column, block and line cutting and pasting from the keyboard or mouse. OS/2 2.0 & 1.3 compatible.

Power, Safety and Speed : Unlimited file sizes, Unlimited UNDO/REDO, and Unlimited number of open files and windows. Complete threaded compiles and error stepping integrated in the editor.

BRIEF Compatible and Configurable: If you use BRIEF you'll feel right at home; a complete BRIEF keyboard compatible layout is included. Powerful keystroke macros and configuration options are available for your own customizations of the editor and keyboard.

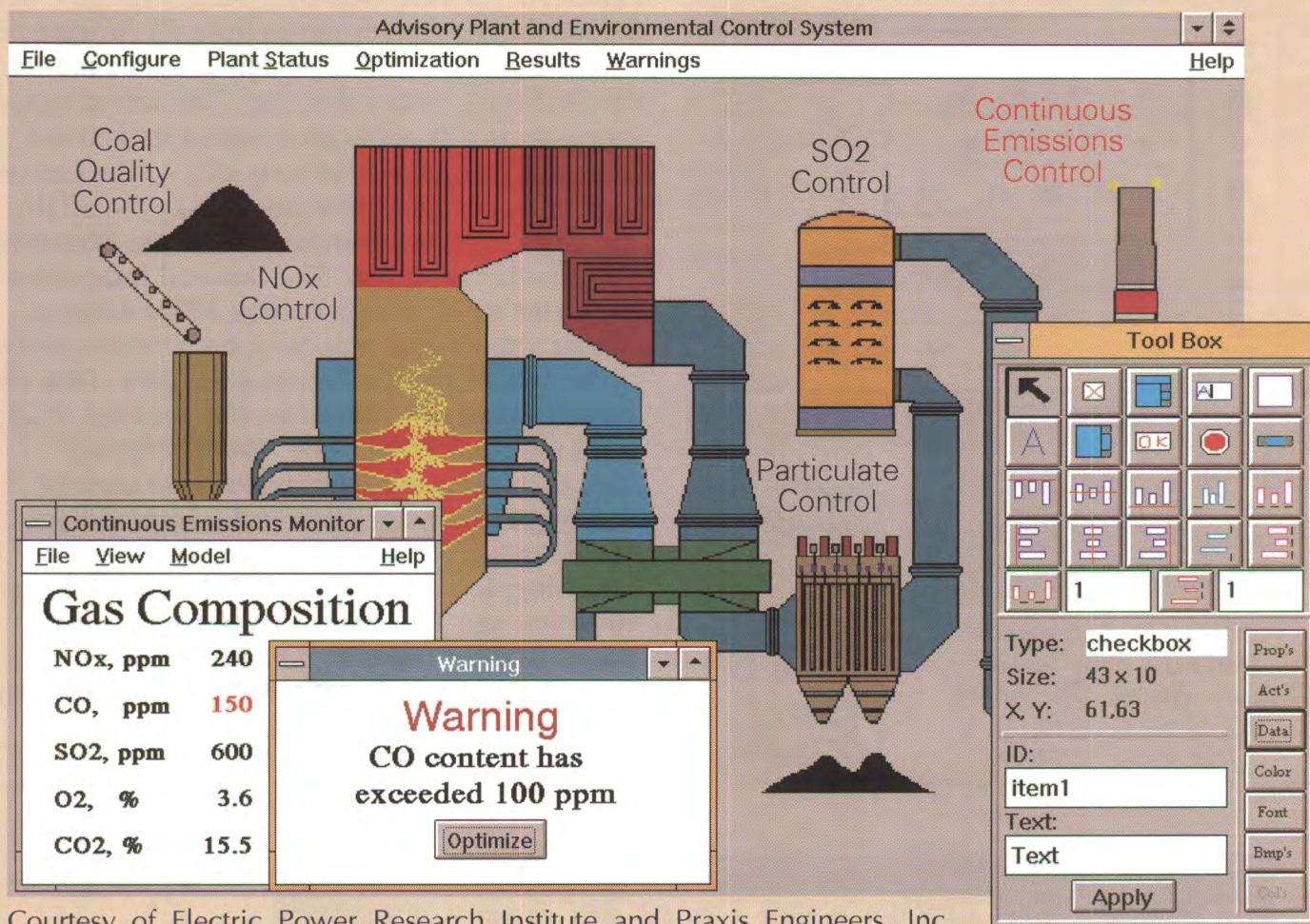
DLL version available: Do you need an editor to embed into your applications? This is perfect for vertical applications and OEMs that do not have the time to reinvent the wheel. Call for details.

Valued Priced: The \$99 special is valid till October 15, 1992. Then normal \$129 list applies.

Then you need to Call ...

RimSTAR Technology 32 Hawkes St. Marblehead, MA 01945. 617-631-7398

GUI Tool for Professional Developers



Courtesy of Electric Power Research Institute and Praxis Engineers, Inc.

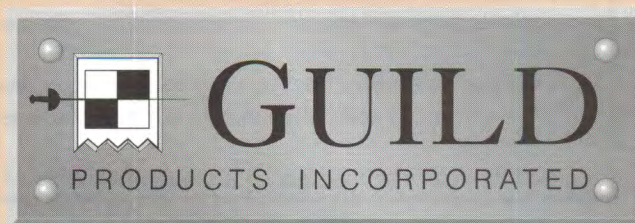
So you'd like to develop serious applications, but you haven't found your ideal GUI tool for OS/2?

GUILD is for YOU!

- **Productive.** Point & click screens, actions and data.
- **Flexible.** Open architecture for custom C/C++ coding.
- **Portable.** Port across Windows, OS/2, Motif and Mac.

(415) 593-3200
Phone

(415) 595-8158
FAX



1301 Shoreway Road, Belmont, Calif. 94002

Trademarks and registered trademarks belong to their respective holders.

Introducing PM/FOCUS



The Power of Graphical Development In an Exciting New Environment

Over three years of development and testing have transformed a vision into reality. An ideal development environment created by the perfect harmony of proven database technology within an exciting new visual landscape. Information Builders introduces PM/FOCUS, a graphical development and decision support system combining unparalleled data access and reporting with the versatility, stability and power of OS/2 2.0.

PM/FOCUS offers a rich graphical tool set complete with File Painter, Forms Painter, and Query Painter to build even the most complex applications without writing code. Application components such as databases, forms, and procedures, become simple graphic objects that can be manipulated and activated with a point, click, and drag of your mouse. The comprehensive array of list boxes, check boxes, multiple fonts, radio buttons and other graphical

images, offer an intuitive interface that's remarkably attractive and easy to use. PM/FOCUS comes with its own proven database engine. Optional interfaces let you create sensational GUI applications for most other popular SQL databases. And PM/FOCUS is EDA/SQL enabled. This means you can have access to over 50 relational and non-relational file types on more than 35 different operating platforms.

PM/FOCUS. It could well be the reason why OS/2 was invented. Call your local Information Builders branch office or Micro Products Telesales at **1-800-969-INFO.** FAX (212) 695-3247.

IBI PM/FOCUS
Information Builders, Inc.
1250 Broadway, New York, NY 10001

Let Gpf write the GUI you design



SYSTEMS, INC.

Using the powerful point and click visual programming environment of Gpf*, you can prototype, test and generate a complete OS/2 PM GUI in a few hours or days rather than the weeks or months required to hand code the same design. Even a relatively simple GUI can require writing thousands of lines of code, but with Gpf you simply draw your user interface on the screen. The integrated dialogue editor of Gpf permits actions and context sensitive help to be linked to controls as you create them. Gpf then generates error free ANSI C complete with embedded SQL statements.

Gpf is optimized to take full advantage of OS/2 PM, the most powerful and robust GUI system available. Since Gpf code directly accesses the PM API, there is no run time module to distribute with your application and no added overhead or royalties.

Gpf keeps the entire design definition in one file. This means single point maintenance for easy update and archiving. From this file Gpf generates the C source file as well as .H, .RC, .IPF, DEF, .IDS, .MAK, etc..

Gpf supports:

- Simple and direct linkage of the interface to program logic, built in or user defined functions.
- Direct association of help screens with controls, and complete integration into the PM Help Presentation Facility.
- Flexible use of Presentation objects (fonts, colors, etc.) with controls and windows (client area and frame).
- Simple inclusion of bitmaps for use on About screens, user-defined buttons, and menu or pulldown entries.
- Automatic embedded SQL statements to read OS/2 DataBase Manager tables, directly into combo or list boxes.
- Multi-thread programming.
- Multiple source file generation.
- Automatic creation of controls that scale with window size.
- Inclusion of user defined controls

Try us out.

Order Gpf today for just \$995.⁰⁰

Call **Gpf Systems Inc.** at:

(203) 873-3300 or (800) 831-0017 - fax (203) 873-3302.

Free demo software available

P.O. Box 414, 30 Falls Rd., Moodus, CT 06469

* GUI Programming Facility